AFRL-RI-RS-TR-2014-103

COGNITIVE JOINTLY OPTIMAL CODE-DIVISION CHANNELIZATION AND ROUTING OVER COOPERATIVE LINKS

STATE UNIVERSITY OF NEW YORK (SUNY) AT BUFFALO

APRIL 2014

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TR-2014-103 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/ S /	**/ S /**
MICHAEL J. MEDLEY	MARK H. LINDERMAN
Work Unit Manager	Technical Advisor, Computing & Communications Division
	Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
APRIL 2014	FINAL TECHNICAL REPORT	OCT 2010 – SEP 2013

4. TITLE AND SUBTITLE

COGNITIVE JOINTLY OPTIMAL CODE-DIVISION CHANNELIZATION AND ROUTING OVER COOPERATIVE LINKS

5a. CONTRACT NUMBER N/A

5b. GRANT NUMBER FA8750-11-1-0016

5c. PROGRAM ELEMENT NUMBER 62702F

6. AUTHOR(S)

Dimitris Pados, Tommaso Melodia, Stella Batalama

5d. PROJECT NUMBER AN11

5e. TASK NUMBER UB

5f. WORK UNIT NUMBER TM

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

State University of New York (SUNY) at Buffalo
501 Capen Hall
Buffalo, NY 14260

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RITE
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER AFRL-RI-RS-TR-2014-103

12. DISTRIBUTION AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

We developed a new spread-spectrum management paradigm in which, unlike mainstream dynamic spectrum access research, digital waveforms are (i) designed to occupy the entire available spectrum and (ii) adaptively track the interference profile at the receiver to maximize link capacity while avoiding interference to existing users. In this context, we studied the problem of maximizing the network throughput of a multi-hop network through joint routing and spread-spectrum channelization. We first presented a centralized formulation of the network control problem. We then developed an algorithm that can be seen as a distributed localized approximation of the throughput-maximizing policy. We refer to the proposed jointly-designed routing and code-division channelization algorithm as ROCH (Routing and cOde-division CHannelization).

15. SUBJECT TERMS

Cooperative communications, hybrid automatic-repeat-request protocols, optimum power and time allocation, outage probability, relaying protocols, wireless networks.

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
U	U	U	SAR	68	**MICHAEL J. MEDLEY**
					19b. TELEPHONE NUMBER (*Include area code*) **N/A**

Contents

I. Summary **1**

II. Introduction **3**

- A. All-spectrum Cognitive Networking through Joint Distributed Channelization and Routing . . . 3
- B. Related Work . . . 5
- C. System Model . . . 7
- D. Physical Layer Model . . . 8

III. Methods, Assumptions, and Procedures **10**

- A. Problem Formulation . . . 10
- B. ROCH: Distributed Joint Routing and Code-Division Channelization . . . 12
- C. Cognitive Secondary Power and Spreading Code Allocation . . . 14
- D. Cognitive Spread-spectrum Channelization Formulation . . . 14
- E. Cognitive Secondary Channel Design . . . 15

IV. Results and Discussion **22**

- A. Performance of Secondary Dynamic Spectrum Access . . . 22
- B. Network Performance Evaluation . . . 22
- C. RcUBe: Real-Time Reconfigurable Radio Framework with Self-Optimization Capabilities . . . 26
- D. Related Work . . . 28
- E. RcUBe Architecture . . . 29
- F. Decision Plane . . . 30
- G. Control Plane . . . 31
- H. Data Plane . . . 33
- I. Register Plane . . . 35
- J. Architectural Considerations . . . 35
- K. Example Design Implementation . . . 36
- L. Experimental Evaluation . . . 38
- M. Software Defined Radio All-spectrum Cognitive Channelization . 44
- N. System Model . . . 45
- O. Optimal Waveform Design . . . 46
- P. Joint Subspace-based Channel and CFO Estimation . . . 47
- Q. Experimental Testbed Setup . . . 50
- R. USRP2 Receiver: Low-cost, Commercial SDRs . . . 50
- S. Transmitter Design . . . 52

V. Conclusions **53**

References **54**

List of Acronyms **62**

List of Figures

Fig. 1: Comparison between code-division channelization and FDM.

Fig. 2: Secondary receiver SINR as a function of the iteration step of the linearized optimizer initialized at the best feasible sample out of $P = 20$ drawings from the $\mathcal{N}(\mathbf{0}, \mathbf{X}'')$ pdf..

Fig. 3: (a) Secondary transmission percentage as a function of the number of active links under Cases $rank(\mathbf{X}'') = 1$ and > 1 (the study includes also the random code assignment scheme); (b) Instantaneous output SINR of a primary signal against primary SINR-QoS threshold SINR_{PU}^{th} (thick line) and instantaneous output SINR of secondary signal; (c) Throughput vs. Number Active Sessions.

Fig. 4: (a) Throughput vs. Number Active Sessions; (b) Throughput vs. Traffic Load; (c) Delay vs. Number Active Sessions.

Fig. 5: RcUBe Architecture.

Fig. 6: Sample FSM architecture taking place in the MAC execution engine at the control plane. State transitions are based on the triplet (events, conditions, actions).

Fig. 7: Routing execution engine for ROSA.

Fig. 8: A view from our testbed implementation.

Fig. 9: Testbed Results for 2-Node Setup.

Fig. 10: Testbed Results for 5-Node Setup with 1-Session at [N1-N5].

Fig. 11: Best SINR values for links in the 5-Node Setup with 1-Session at [N1-N5].

Fig. 12: USRP2 block diagram with attached RFX2400.

Fig. 13: Secondary transmitter block diagram.

Fig. 14: Secondary receiver block diagram.

List of Tables

TABLE I (page 42) : List of *Events*, *Conditions*, and *Actions*.

I. Summary

We consider a secondary multi-hop cognitive radio network with decentralized control that operates cognitively to coexist with primary users. We propose a new spread-spectrum management paradigm, in which, unlike mainstream dynamic spectrum access research, digital waveforms are designed to occupy the entire available spectrum, and to adaptively track the interference profile at the receiver to maximize the link capacity while avoiding interference to primary users. In this context, we study the problem of maximizing the network throughput of a multi-hop network through joint routing and spread-spectrum channelization. We first propose a centralized formulation of the network control problem. We then propose an algorithm that can be seen as a distributed localized approximation of the throughput-maximizing policy. We refer to the proposed jointly-designed routing and code-division channelization algorithm as ROCH (Routing and cOde-division CHannelization). Specifically, power and spreading code are jointly selected to maximize the pre-detection secondary SINR while providing quality of service guarantees to on-going primary and secondary transmissions, while the routing algorithm dynamically selects relays based on the network traffic dynamics and on the achievable data rates on different secondary links. We study the throughput and delay performance of ROCH through a extensive simulation experiments, which demonstrate the appeal of the proposed framework through significant performance gains compared to baseline solutions.

Existing commercial wireless systems are mostly hardware-based, and rely on closed and mostly inflexible designs and architectures. Moreover, while in recent years there have been significant algorithmic developments in cross-layer network adaptation and resource allocation, in most cases existing architectures are not flexible enough to incorporate these advancements other than through ad hoc tweaks. To address these limitations, we propose RcUBe (Real-time Reconfigurable Radio), a novel radio framework that offers real-time reconfigurability and self-optimization capabilities at the PHY, MAC, and network layers of the protocol stack. Unlike state-of-the-art solutions designed to provide flexibility at a specific layer of the protocol stack, RcUBe is a flexible architectural abstraction spanning the three lowest layers of the protocol stack. RcUBe provides abstractions to define in a structured way sophisticated decision-making algorithms and applying the corresponding reconfigurations in real-time. RcUBe provides these features through a design structured into four planes, namely *decision*, *control*, *data*, and *register* plane. The decision plane consists of user-defined decision algorithms that provide the control logic for real-time protocol optimization, adaptation, and reconfiguration. In the control plane, RcUBe stores the required logic for routing and MAC protocol execution as well as data plane management. Data processing takes place in the data plane, while the register plane stores and manages access to system parameters and environmental information. We illustrate the unique features and the flexible design of RcUBe by implementing a cross-layer architecture that performs on-the-fly

reconfigurations based upon decisions taken in real time by a decentralized optimization algorithm.

Finally, we design and implement a wireless software defined radio (SDR) testbed for cognitive channelization in multipath fading environments. Primary and secondary users coexist and utilize the same spectrum concurrently. Cognitive channelization is achieved by adaptive waveform assignment to secondary users that assumes unknown primary users and maximizes the signal-to-interference-plus-noise ratio (SINR) at the output of the secondary receiver, while at the same time maintains interference to primary users below a prespecified threshold. We deployed four commercial, low-cost SDR transceivers in an indoors lab environment. To address challenges related to low-cost, off-the-shelf testbed components and in particular to unstable clock oscillators that introduce carrier frequency offsets (CFOs) between any transmitter-receiver pair, we propose a subspace based technique for joint blind estimation of the multipath channel coefficients and the introduced CFOs. The theoretical concepts of adaptive cognitive channelization in the presence of unknown primary users and multipath fading are experimentally demonstrated and validated in terms of instantaneous SINR and bit-error-rate (BER).

II. Introduction

A. All-spectrum Cognitive Networking through Joint Distributed Channelization and Routing

Cognitive radio networks [1]-[2] have emerged as a promising technology to improve the utilization efficiency of the existing radio spectrum. Mainstream cognitive radio proposals focus on opportunistic access to the licensed spectrum where the primary users of the band are known a priori and this knowledge can be utilized to detect if the band is occupied by the known signal pattern. Quite the opposite, in the unlicensed band there are potentially many uncoordinated devices, and their signal waveforms and activation statistics are in general unknown. Moreover, in cognitive radio networks with multi-hop communication requirements, spectrum occupancy is location-dependent, and the receiver interference profile may, thus, vary at each relay node.

We therefore propose a new spread-spectrum management paradigm, in which waveforms are designed to occupy the entire available spectrum without generating harmful interference to active primary or secondary users. In this way, the secondary users share the licensed spectrum with the primary users to achieve frequency reuse. At the same time, the dynamic and location-dependent nature of the wireless environment calls for the development of routing algorithms that are aware of the interference profile at each potential relay.

The lack of established infrastructure and the wireless channel dynamics impose an unprecedented set of challenges over spread-spectrum cognitive ad hoc networks. First, secondary users should optimize the spreading code and power to avoid generating harmful interference to primary users. The challenges here arise from the assumption that the spreading codes of primary users are unknown to the secondary users. Second, in a multi-hop network, the spectrum environment varies in time and space depending on the activities of primary users, interference, and fading. The optimal spectrum-spreading channelization may therefore be different at each hop in a multi-hop path. Furthermore, as new secondary links are formed and others vanish, and following network traffic dynamics, routing of data flows from one secondary node to another may frequently change. Therefore, controlling the interaction between routing and code design is of fundamental importance.

In this work, we explore a new framework that captures the interdependencies between spread-spectrum channelization and routing. The throughput optimization is carried out dynamically by all secondary transmitters to continuously adapt to the changing spectrum environment and traffic arrival rates. Specifically, a distributed algorithm for dynamic joint power and spread-spectrum channelization is developed to maximize the pre-detection secondary SINR while guaranteeing the SINR-QoS requirements for on-going transmissions from primary and secondary users. The excellent cognitive network performance characteristics is demonstrated by simulation studies included in this project.

A motivating example. Through an example, we will try to intuitively highlight the potential benefits of spread-spectrum channelization as compared to traditional frequency-division multiplexing (FDM) for dynamic spectrum access. In code-division channelization, secondary users use spread spectrum signals that occupy the whole spectrum band. In this example, $16-bit$ spreading codes over the real field are assigned to secondary users. In FDM, the whole spectrum band is divided into 16 subbands, and each primary user is assigned with a specific subband. In FDM scenario, secondary users can access a single subband only if the subband is not occupied by primary users or any other secondary users. In the simulation, the transmission SNRs of primary links are all set equal to $15\,dB$; the maximum allowable transmission SNR for secondary links is set to $15\,dB$. All the signature vectors for primary links are generated from a minimum total-squared correlation optimal binary signature set which achieves the Karystinos-Pados (KP) bound [32]-[34] for each $(K, 16)$ pair of values[1], where K is the number of active primary users. Both of code-division channelization and FDM schemes provide the same minimum SINR guarantee $3\,dB$ for both active primary users and secondary users.

Note that we require that both the code-division channelization and the FDM schemes provide the same minimum SINR guarantee for both active primary users and secondary users. We vary the number of active primary users and evaluate the maximum number of admissions for secondary users with the two different schemes. As shown in Fig. 1, it can be observed that spread spectrum management with code-division channelization allows more secondary users to simultaneously access the spectrum than FDM spectrum management. For example, with 16 subbands and 14 active primary users, only 2 secondary users can access the network with FDM. While up to 7 secondary users can access the network with carefully designed code-division channelization.

Contributions. Within this context, our main contributions in this project can be outlined as follows:

- **Uncoordinated code-division spectrum management.** Unlike mainstream work on cognitive radio networks, we consider a code-division channelization based secondary network that operates cognitively to coexist with primary users for infrastructure-less cognitive radio ad hoc networks.

- **Distributed joint routing and code-division channelization**. We formulate a joint routing and code-division channelization problem. Given the centralized nature and high computational complexity of the problem, we study localized algorithms for joint dynamic routing and spread-spectrum channelization that are designed to maximize the global objective function of the centralized problem. To the best of our knowledge, no existing algorithm attempts to control the routing and code-division channelization functionalities to jointly maximize the network throughput.

- **Novel low-complexity spread-spectrum channelization algorithms.** We study the problem of designing a joint power and spreading code al-

[1]When $K <= 16$, the KP-optimal sequences coincide with the familiar Walsh-Hadamard signature codes.

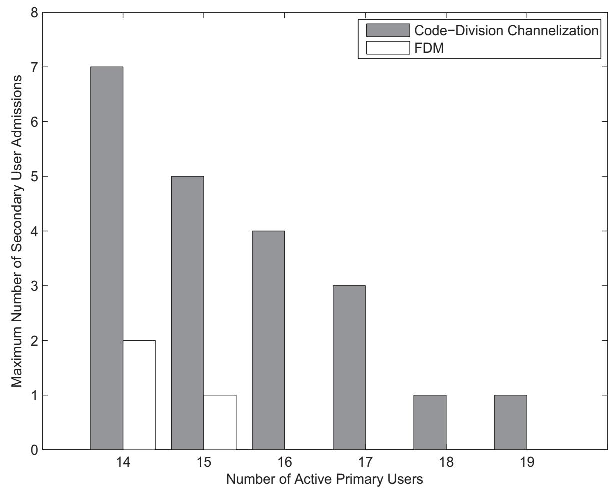


Fig. 1: Comparison between code-division channelization and FDM.

location scheme for the secondary link that maximizes the pre-detection SINR under quality of service constraints for the secondary users. Using semidefinite relaxation, we develop a novel low-complexity suboptimal solution to an otherwise NP-hard problem with excellent performance in practice.

B. Related Work

Since spectrum occupancy is location-dependent, the spectrum occupancy profile may be different at each relay node in a multi-hop end-to-end path. Therefore, one of the key challenges in the design of cognitive ad hoc networks is to jointly and dynamically allocate routes and portions of the spectrum to each node in the multi-hop network. The authors in [3] proposed a routing algorithm based on a probabilistic estimation of the available capacity of each secondary link. The proposed probability-based routing metric relies on the probability distribution of the interference between primary and secondary users over a given channel. In [4, 5], the authors proposed a connectivity-driven routing algorithm, where paths are measured in terms of their degree of connectivity in a multi-hop cognitive radio network that is highly influenced by the primary user behavior. Route-stability-oriented routing is introduced in [6] based on the concept of route maintenance cost. The maintenance cost represents the effort needed or penalty paid to maintaining end-to-end connectivity. In [7], a distributed and localized algorithm for joint dynamic routing and spectrum allocation for ad hoc cognitive radio networks is proposed. The proposed algorithm jointly addresses routing and spectrum assignment with power control under the physical interference model. In [8], a coordination scheme is introduced that allows secondary users to coordinate with primary users. Specifically, secondary users offer their services as an intermediate relay node in an effort to improve throughput of primary users utilizing a 802.11-based channel access mechanism. In return, the secondary user 'piggy-backs' some of its own data while acting as a relay. In [9], the authors propose the CRP routing protocol that considers joint spectrum-route selection and service differentiation in CR routes. The proposed protocol allows for two classes of routes - class I routes that provide better CR network performance, while class II routes aim to achieve a higher measure of protection for the PUs. The authors in [10] proposed a distributed routing algorithm where secondary users minimize their interference to the primary users while keeping the delay along the route low. The reader is referred to [11] and references therein for an excellent survey of the main results in this area.

Herein, we consider cognitive networks built around a primary code-division multiplexing (CDM) system. Unlike traditional frequency division operations where cognitive secondary users may transmit opportunistically in sensed spectrum holes/void only, cognitive code-division users may operate in parallel in frequency and time to a primary system as long as the induced spread-spectrum interference remains below a pre-defined acceptable threshold[2]. Power control under an "interference temperature" constraint (total secondary user disturbance power over primary band) was considered [12] in cognitive code-division systems. Furthermore, a joint power allocation and admission control algorithm was proposed in [13] for cognitive CDMA systems to maximize the energy efficiency of secondary users with QoS guarantee for primary and secondary links. In [14], a spectrum underlay (CDM-based) cognitive radio network (CRN) was considered and a low-complexity suboptimal algorithm, named interference constraint-aware stepwise maximum interference removal algorithm (I-SMIRA), was proposed to maximize the number of admitted secondary users by optimizing power and transmission rate. However, no code-channel (signature) optimization was carried out for the secondary users. In contrast, in [15] a secondary code assignment scheme was presented to obtain the binary secondary signature by hard-limiting the code sequence that exhibits the minimum mean-square cross-correlation with the primary received signal. The secondary code set of multiple secondary users was also constructed in an iterative way. Under interference-minimizing code assignments, bit rate and spreading factor adjustments for a secondary CDMA system were considered in [16]. A distributed algorithm for resource allocation of spectral bands, power, and data rates among multiple secondary users for multi-carrier CDMA systems was developed in [17]. The problem of beamforming and power control for CR networks aimed at restricting the interference on PUs has been addressed in [18]. The authors in [19] study the problem of fair spectrum sharing among all SUs in underlay CR

[2]While early standardization and regulation discussions have begun [23], no conclusive "interference temperature" rules and agreements have been reached yet.

networks, subject to certain QoS (in terms of minimum SINR and transmission rate) and outage probability of constraints with imperfect channel state information. A joint admission control and rate/power allocation for secondary users is proposed. In [20], the authors consider the scenario where SUs can adaptively adjust their transmit directions in addition to transmit power according to available channel information. A joint power allocation and phase control solution is proposed subject to interference and power constraints. In [21], the authors study the optimal scheduling problem with the objective to achieve proportional fairness of the long-term average transmission rates among different links in a cognitive ad hoc network with spectrum underlay. Hybrid overlay/underlay CR waveforms were designed to adapt its spectrum to efficiently exploit both unused spectrum holes and underused spectrum bands through OFDM and MC-CDMA in [22].

The common theme of most research in the area of CDMA-based CR networks is to optimize physical layer performance, without modeling in detail how physical layer resource allocation interacts with higher layers of the protocol stack to improve network performance metrics. In comparison with work in this area, our work herein exhibits the following novelties: 1) different from [12, 13, 14], which consider fixed spreading code assignment, we introduce one additional degree of freedom, i.e., we attempt to optimize the spreading codes at the physical layer to further improve the system performance. We develop a novel low-complexity suboptimal solution to an otherwise NP-hard problem with excellent performance in practice; 2) in our work, interference between secondary links is taken into consideration, and the SINR requirement necessary to guarantee a certain level of quality of service for secondary users is also guaranteed. This is different from [15] and [16], where only the interference from secondary to primary users is considered; 3) direct sequence spread spectrum technology is adopted in our work, while [17] and [22] consider CDMA-based multi-carrier modulation; 4) no previous work [12] - [22] considers interactions between the higher layer functionalities of the networking protocol stack (i.e., routing and scheduling) and physical layer resource allocation (i.e., power and spreading code design) in a multihop cognitive radio network scenario. In our work, we explore a new framework that captures the interdependencies between spread-spectrum channelization and routing in cognitive radio ad hoc networks. To the best of our knowledge, our work is the first to consider joint code-channel optimization and routing in cognitive ad hoc networks.

C. System Model

We consider a primary spread-spectrum system with processing gain (code sequence length) G. Denote $\mathcal{PU}$ as the set of active primary communication links. Let (l,k) denote the link with transmitter l and receiver k. Note that link (l,k) is distinct from link (k,l). Each primary link is pre-assigned with an unique code sequence, i.e., $\mathbf{s}_{lk}$ for link (k,l). We let $\mathcal{N} = \{1, \cdots, N\}$ represent a finite set of secondary users (also referred to as nodes). Secondary users do not have any pre-assigned code sequence and opportunistically send their data by optimizing code sequence and power.

Traffic flows of secondary users are, in general, carried over multi-hop routes. Let the traffic demands consist of a set $\mathcal{D} = \{1, 2, \cdots, D\}$ of unicast sessions. Each session $d \in \mathcal{D}$ is characterized by the destination node $d, d \in \mathcal{N}$ for the traffic. We indicate the arrival rate of session d at node i as $\mu_i^d(t)$ at time t. Each node maintains a separate queue for each session d for which it is either a source or an intermediate relay. At time slot t, define $Q_i^d(t)$ as the number of queued packets of session d waiting for transmission at secondary user i. Define $r_{ij}^d(t)$ (in packets/s) as the transmission rate on link (i, j) for session d during time slot t, and $\mathbf{R}$ as the vector of rates. Note that $\mu_i^d(t)$ represents the exogenous traffic arrivals at node i, while $\sum_{k \in \mathcal{N}, k \neq i} r_{ki}^d(t)$ represents the total endogenous traffic arrivals at node i resulting from routing and transmission decisions from other nodes k.

For $\forall i \in \mathcal{N}$, the queue is updated as follows:

$$Q_i^d(t+1) = \left[Q_i^d(t) + \sum_{k \in \mathcal{N}, k \neq i} r_{ki}^d(t) - \sum_{j \in \mathcal{N}, j \neq i} r_{ij}^d(t) + \mu_i^d(t) \right]^+ .$$

D. Physical Layer Model

We recall that the secondary cognitive ad hoc network coexists with the primary system over the primary licensed band. In general, the transmitted spread spectrum signal for the link (i, j) is denoted by

$$u_{ij}(t) = \sum_{m=1}^{\infty} \sqrt{E_i} s_{ij} (t - mT) e^{j(2\pi f_c t + \phi_i)} b_i(m) \tag{1}$$

where $b_i(m) \in \{-1, +1\}$ is the mth data bit (binary phase-shift-keying data modulation), E_i is the total transmission energy, and ϕ_i is the carrier phase with carrier frequency f_c; $s_{ij}(t)$ is the normalized unit-energy user signature waveform with duration T given by

$$s_{ij}(t) = \sum_{g=0}^{G-1} \mathbf{s}_{ij}(g) \psi(t - gT_c) \tag{2}$$

where $\mathbf{s}_{ij}(g),\ g = 0, 1, \ldots, G-1$, is the value of the gth chip of the spreading-code vector of the link (i, j), $\psi(t)$ is the chip waveform, and $T_c = \frac{T}{G}$ is the chip period. The combined received signal waveform due to all link transmissions over flat fading channels of impulse response $h_{ij}(t)$

$$y(t) = \sum_{(i,j) \in \mathcal{PU} \cup \mathcal{SU}} h_{ij}(t) u_{ij}(t - \tau_{ij}) + n(t) \tag{3}$$

where $\mathcal{SU}$ is the set of active secondary links, $\tau_{ij} \in [0, T)$ is the relative delay of link (i, j) with respect to the link of interest, and $n(t)$ is a white Gaussian noise process. After carrier demodulation, chip-matched filtering and sampling at the chip rate over the duration of a symbol (bit) period of G chips, the received signal at primary node k over the link of interest (l,k), denoted as $\mathbf{y}_{lk}^{(p)}$, can be represented as

$$\mathbf{y}_{lk}^{(p)} = \sqrt{E_l} L_{lk} \mathbf{s}_{lk} b_l + \sum_{\substack{(m,n)\in\mathcal{PU} \\ (m,n)\neq(l,k)}} \sqrt{E_m} L_{mk} \mathbf{s}_{mn} b_m + \sum_{(u,v)\in\mathcal{SU}} \sqrt{E_u} L_{uk} \mathbf{c}_{uv} b_u + \mathbf{n}_k. \tag{4}$$

We note that the superscript "p" in (4) denotes primary node. Similarly, let $\mathbf{y}_{ij}^{(s)}$ denote the secondary signal received at node j over the link of interest (i,j) ,

$$\mathbf{y}_{ij}^{(s)} = \sqrt{E_i} L_{ij} \mathbf{c}_{ij} b_i + \sum_{(m,n)\in\mathcal{PU}} \sqrt{E_m} L_{mj} \mathbf{s}_{mn} b_m + \sum_{\substack{(u,v)\in\mathcal{SU} \\ (u,v)\neq(i,j)}} \sqrt{E_u} L_{uj} \mathbf{c}_{uv} b_u + \mathbf{n}_j, \tag{5}$$

where $E_l > 0$, $b_l \in \{\pm 1\}$, and $\mathbf{s}_{lk} \in \mathbb{R}^G$, $\|\mathbf{s}_{lk}\| = 1$, denote bit energy, information bit, and normalized signature vector of primary user l over primary link (l,k), $(l,k) \in \mathcal{PU}$, respectively; $E_i > 0$, $b_i \in \{\pm 1\}$, and $\mathbf{c}_{ij} \in \mathbb{R}^G$, $\|\mathbf{c}_{ij}\| = 1$, denote the bit energy, information bit, and normalized signature vector, respectively, of secondary user i over secondary link (i,j), $(i,j) \in \mathcal{SU}$; L_{lk}, $(l,k) \in \mathcal{PU}$ and L_{ij}, $(i,j) \in \mathcal{SU}$ denote the path coefficients for primary link (l,k) and secondary link (i,j), respectively. $\mathbf{n}_k$ and $\mathbf{n}_j$ represent additive white Gaussian noise (AWGN) at primary node k and secondary node j, correspondingly, independent from each other with $\mathbf{0}$ mean and autocovariance matrix $\sigma^2\mathbf{I}$. We note that the superscript "s" in (5) denotes secondary node.

Rather than being fixed, the interference between the secondary and primary links varies with the transmit power and spreading code allocation. As modeled in (4) and (5), interference from both primary links and secondary links is considered in our design. Specifically, as shown in (4), which is the received signal at the receiver node k of primary link (l,k), the second term $\sum_{\substack{(m,n)\in\mathcal{PU} \\ (m,n)\neq(l,k)}} \sqrt{E_m} L_{mk} \mathbf{s}_{mn} b_m$ represents the interference from other primary links, and the third term $\sum_{(u,v)\in\mathcal{SU}} \sqrt{E_u} L_{uk} \mathbf{c}_{uv} b_u$ represents the interference from secondary links. Similarly, in (5), which is the received signal at receiver node j of secondary link (i,j), the second term $\sum_{(m,n)\in\mathcal{PU}} \sqrt{E_m} L_{mj} \mathbf{s}_{mn} b_m$ represents the interference from primary links, and the third term $\sum_{\substack{(u,v)\in\mathcal{SU} \\ (u,v)\neq(i,j)}} \sqrt{E_u} L_{uj} \mathbf{c}_{uv} b_u$ represents the interference from other secondary links.

The linear filters at the primary and secondary receivers that exhibit maximum output SINR [24] can be found to be

$$\mathbf{w}_{max\mathrm{SINR},lk} = c_1 \mathbf{A}_{lk}^{-1} \mathbf{s}_{lk}, \ \ (l,k) \in \mathcal{PU},$$
$$\mathbf{w}_{max\mathrm{SINR},ij} = c_2 \mathbf{A}_{ij}^{-1} \mathbf{c}_{ij}, \ \ (i,j) \in \mathcal{SU},$$

where $\mathbf{A}_{lk} = \mathbb{E}\{\mathbf{y}_{lk}^{(p)} \mathbf{y}_{lk}^{(p)T}\}$, $\mathbf{A}_{ij} = \mathbb{E}\{\mathbf{y}_{ij}^{(s)} \mathbf{y}_{ij}^{(s)T}\}$, c_1 and c_2 are arbitrary positive constants, i.e., $c_1, c_2 > 0$. Maximum output SINRs at the receiver end for link (l,k) and (i,j) are, respectively, given by

$$\mathrm{SINR}_{lk} = L_{lk}^2 E_l \mathbf{s}_{lk}^T \mathbf{A}_{\backslash(l,k)}^{-1} \mathbf{s}_{lk}, \ \ \forall (l,k) \in \mathcal{PU},$$
$$\mathrm{SINR}_{ij} = L_{ij}^2 E_i \mathbf{s}_{ij}^T \mathbf{A}_{\backslash(i,j)}^{-1} \mathbf{s}_{ij}, \ \ \forall (i,j) \in \mathcal{SU},$$

where $\mathbf{A}_{\backslash(l,k)}$ and $\mathbf{A}_{\backslash(i,j)}$ are the disturbance autocorrelation matrices at the receiver end for link (l,k) and (i,j), respectively, defined by

$$\mathbf{A}_{\backslash(l,k)} \triangleq \sum_{\substack{(m,n)\in\mathcal{PU} \\ (m,n)\neq(l,k)}} L_{mk}^2 E_m \mathbf{s}_{mn} \mathbf{s}_{mn}^T + \sum_{(u,v)\in\mathcal{SU}} L_{uk}^2 E_u \mathbf{c}_{uv} \mathbf{c}_{uv}^T + \sigma^2 \mathbf{I},$$
$$\mathbf{A}_{\backslash(i,j)} \triangleq \sum_{(m,n)\in\mathcal{PU}} L_{mj}^2 E_m \mathbf{s}_{mn} \mathbf{s}_{mn}^T + \sum_{\substack{(u,v)\in\mathcal{SU} \\ (u,v)\neq(i,j)}} L_{uj}^2 E_u \mathbf{c}_{uv} \mathbf{c}_{uv}^T + \sigma^2 \mathbf{I}.$$

In our cognitive radio setup, the normalized channel capacity $\frac{C_{ij}}{W}$ of secondary link (i,j) as a function of SINR_{ij} is given by [25]

$$\frac{C_{ij}}{W} = \log_2(1 + \frac{C_{ij}}{W} SINR_{ij}) \tag{6}$$

where C_{ij} is the channel capacity and W is the bandwidth of the primary licensed band. Given the fixed bandwidth W, the channel capacity C_{ij} can be evaluated by (6) for any instantaneous value of SINR_{ij}.

III. Methods, Assumptions, and Procedures

A. Problem Formulation

Our goal is to design a distributed cross-layer control algorithm to maximize the secondary network throughput by jointly and dynamically allocating routes, code sequence and transmit power for each secondary link along the path. Denote $\mathcal{SU}(t)$ as the set of secondary links chosen for activation during time slot t, $\mathbf{c}(t) = \{\mathbf{c}_{ij}(t) : (i,j) \in \mathcal{SU}(t)\}$ and $\mathbf{E}(t) = \{\mathbf{E}_i(t) : (i,j) \in \mathcal{SU}(t)\}$ as the sets of code sequences and power allocation decisions for every active secondary link. An ideal throughput-optimal network controller should, at each decision period (i.e. time slot t), find $\mathcal{SU}(t)$, $\mathbf{c}(t)$, and $\mathbf{E}(t)$ to maximize

$$\sum_{i\in\mathcal{SU}} \sum_{j\in\mathcal{SU}, j\neq i} C_{ij}(\mathbf{c}(t), \mathbf{E}(t)) \cdot \Delta Q_{ij}(t), \tag{7}$$

where $\Delta Q_{ij}(t) = \max_{d \in \mathcal{D}} \left[Q_i^d(t) - Q_j^d(t) \right]^+$. The transmission rates are then given by

$$r_{ij}^d(t) = \begin{cases} C_{ij}(\mathbf{c}(t), \mathbf{E}(t)) & \text{if } d = d_{ij}^*(t) \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where $d_{ij}^*(t) = \arg\max_{d \in \mathcal{D}} \left\{ Q_i^d(t) - Q_j^d(t) \right\}$.

The objective function (7) is defined based on the principle of dynamic back-pressure, first introduced in [26]. It can be proven [27] that a control strategy that jointly assigns resources at the physical/link layers and routes to maximize the weighted sum of differential backlogs (with weights given by the achievable data rates on the link) achieves throughput optimality, in the sense that it is able to keep all network queues finite for any level of offered traffic within the network capacity region. Moreover, a desirable solution should enable secondary users to dynamically utilize the available spectrum resource in the code domain to provide SINR guarantees for both primary and secondary users. The problem can be expressed as

$$\begin{aligned}
&\mathbf{P1} : \mathit{Find} : && \mathcal{SU}(t),\ \mathbf{E}(t),\ \mathbf{c}(t) \\
&\mathit{Maximize} : && \sum_{i \in \mathcal{SU}} \sum_{j \in \mathcal{SU}} C_{ij}(\mathbf{c}(t), \mathbf{E}(t)) \cdot \Delta Q_{ij}(t) \\
&\mathit{Subject\ to} : && \\
& && \mathrm{SINR}_{lk} \geq \mathrm{SINR}_{PU}^{th}, \quad \forall (l,k) \in \mathcal{PU}, \\
& && \mathrm{SINR}_{ij} \geq \mathrm{SINR}_{SU}^{th}, \quad \forall (i,j) \in \mathcal{SU}(t), \\
& && \mathbf{c}_{ij}(t)^T \mathbf{c}_{ij}(t) = 1,\ E_i \leq E_{max}, \quad \forall (i,j) \in \mathcal{SU}(t).
\end{aligned}$$

Therefore, ideally, a throughput-optimal policy should continuously (i.e., at each time slot) assign resources on each network link by solving problem **P1** to optimality. However, exact solution of **P1** requires global knowledge of all feasible allocations and a centralized algorithm to solve a mixed integer non-linear problem (NP-hard in general) such as **P1** on a time-slot basis. This is clearly not practical for real-time decision making. This provides the rationale for our distributed algorithm, which is designed to provide an approximate solution to **P1** based on real-time distributed decisions driven by locally collected information. Note that, for the sake of simplicity, we will drop all time dependencies.

In the following sections, we first discuss the decentralized solution for joint routing and code-division channelization in Section V. Specifically, we described in detail about how next hops are selected together with physical layer resource optimization in our proposed algorithm. We also discussed how nodes learn about the environment to make distributed decisions on routing based on a combination of physical sensing and of local exchange of information through control packets at MAC layer. Then, we present in detail our power and spreading code allocation algorithm to maximize the SINR for secondary links in Section VI.

B. ROCH: Distributed Joint Routing and Code-Division Channelization

We now present the decentralized joint routing and code-division channelization solution, which aims at maximizing throughput through joint opportunistic routing, dynamic code sequence optimization and transmit power control. In the proposed solution, backlogged nodes i first maximize their local objective function $C_{ij} \cdot \Delta Q_{ij}$ over all feasible next hops j by optimizing $\mathbf{c}_{ij}$ and E_i based on locally collected spectrum information - details are given in what follows. Then, in case of channel access contention, each node will access the channel with a probability that is a monotonically increasing function of its local utility. We now describe the details of the proposed solution. Every backlogged node i performs the following algorithm:

1. Find the set of feasible next hops $\{n_1^d, n_2^d, \cdots, n_k^d\}$ for the backlogged session d, which are neighbors with positive advance towards the destination of d. We say node j has *positive advance* with respect to i iff j is closer to the destination d than i.

2. For each candidate next hop $j \in \{n_1^d, n_2^d, \cdots, n_k^d\}$, maximize link capacity C_{ij} by optimizing $\mathbf{c}_{ij}$ and E_i, using the algorithm proposed in the following Section VI.

3. Schedule the session d^* with next hop j^* with maximal $C_{ij} \cdot \Delta Q_{ij}$. Hence, routing is performed in such a way that lightly backlogged nodes with higher link capacity receive most of the traffic.

4. Once the next hop and corresponding code sequence and power allocation have been determined, the probability of accessing the medium is calculated based on the value of $C_{ij} \cdot \Delta Q_{ij}$. Nodes with higher $C_{ij} \cdot \Delta Q_{ij}$ will have a higher probability of accessing the medium and transmit. Note that links with higher differential backlog may have higher spectrum utility, and thus have higher probability of being scheduled for transmission. This is implemented by varying the size of the contention window at the MAC layer. The transmitting node i generates a backoff counter BC_i chosen randomly (with a uniform distribution) within the interval $[1, 2^{CW_i - 1}]$, where CW_i is the contention window of transmitter i, whose value is a decreasing function $\Phi()$ of $C_{ij} \cdot \Delta Q_{ij}$ as below

$$CW_i = -\theta_1 \cdot \frac{C_{ij} \cdot \Delta Q_{ij}}{\sum_{k \in \mathcal{N}_i, k, l \in \mathcal{V}} (C_{kl} \cdot \Delta Q_{kl})} + \theta_2, \tag{9}$$

   where $\theta_1 > 0, \theta_2 > 0$ and the denominator represents the objective value of neighboring competing nodes. Note that sender i collects the utility values of its neighbors by overhearing control packets coded by a common spreading code.

Note that only the second step involves solving a real optimization (maximization) problem (with low complexity). Specifically, in step 2, the user needs

Algorithm 1 ROCH Algorithm

1: At backlogged node i
2: **for each** backlogged session d **do**
3:     **for** $j \in \{n_1^d, n_2^d, ..., n_k^d\}$ **do**
4:         Calculate link capacity C_{ij} by optimizing $\mathbf{c}_{ij}$ and E_i using algorithm 2 proposed in the following Section VI.
5:     **end for**
6: **end for**
7: Schedule $(d^*, j^*) = \arg\max\,(C_{ij} \cdot \Delta Q_{ij})$
8: Set contention window $CW_i = -\theta_1 \cdot \frac{C_{ij} \cdot \Delta Q_{ij}}{\sum_{k \in \mathcal{N}_i, k, l \in \mathcal{V}} (C_{kl} \cdot \Delta Q_{kl})} + \theta_2$
9: Return $[s^*, j^*, CW_i]$

to solve the optimal power and spreading code allocation problem by performing the algorithm discussed in Section VI (which has polynomial complexity). In step 3, the user simply selects the next hop (and the associated session) with maximal utility by comparing utility values of all candidate next hops. In step 4, the user simply calculates the contention window size based on (9) for MAC layer.

The algorithm calculates the next hop j opportunistically depending on queueing and spectrum dynamics, according to the objective function $C_{ij} \cdot \Delta Q_{ij}$. The combination of opportunistically selected next hops leads to a multi-hop path. The multi-hop path discovery terminates when the destination is selected as the next hop. If the destination is in the transmission range of the transmitter (either a source or an intermediate hop for that session), the differential backlog between the transmitter and the destination is no less than the differential backlogs between the transmitter and any other nodes, because the queue length of the destination is zero. With this scheme, lightly-congested nodes (as indicated by a smaller differential backlog) have a higher probability of being selected as intermediate relays. Links with larger differential backlogs result in smaller contention window size at the medium access control, and therefore have higher probability of accessing the channel to reserve resources. Ultimately, heavily backlogged nodes with potential high-data rate opportunities to transmit have a higher probability of accessing the channel.

The algorithm is implemented through a MAC protocol that uses a three-way handshake between source and destination. The three-way handshake is carried out via exchange of Request-to-Send (RTS), Clear-to-Send (CTS) and Data Transmission reServation (DTS) packets. Backlogged nodes contend for spectrum access on a common control channel that operates in parallel to the data channel through a common spreading code. Each node makes adaptive decisions based on local information collected through RTS/CTS/DTS packets coded with a common spreading code. The DTS packet is used to announce the information on spreading code, transmit power allocation and queueing information to neighboring nodes. We include queuing information in the control packets to allow the nodes to make routing decisions. The set of feasible next hops can be obtained by requiring only the neighborhood geographical information (can be obtain by GPS for example) and an estimate of the destination's position. Geographical information, together with the queue information, are encapsulated in the control packets (i.e., RTS/CTS/DTS) to allow nodes to exchange their information.

C. Cognitive Secondary Power and Spreading Code Allocation

We now come to the spread-spectrum channelization problem, which is a crucial component of our spectrum management framework and is executed at step 2 of the algorithm described in the previous section. We would like to dynamically and adaptively generate digital waveforms that span the whole available spectrum band to maximize the capacity of the link, and that at the same time avoid interfering with existing users (what we refer to as spread-spectrum channelization). Each secondary user optimizes the waveform to be used based on the current interference condition at the secondary receiver and on the interference receiver profile to maximize the overall network throughput, as described in Section V. We present the secondary code-channel optimization formulation in Section VI-A. The formulated problem is NP-hard with complexity exponential in G. In Section VI-B, we develop a realizable suboptimal solution with polynomial complexity.

D. Cognitive Spread-spectrum Channelization Formulation

In order for a cognitive radio network to efficiently share the licensed band with the primary network, the secondary transmitter has to guarantee the SINR QoS of all primary users and ongoing secondary users and, simultaneously, maximize the secondary SINR of link (i, j) at its receiver end. We first assume that the secondary link (i, j) is activated with the transmission bit energy E_i and (normalized) signature vector $\mathbf{c}_{ij}$. In this spirit, our objective is to find the pair $(E_i, \mathbf{c}_{ij})$ that maximizes SINR_{ij} under the constraints that SINR_{lk}, $\forall(l, k) \in \mathcal{PU}$, and SINR_{uv}, $\forall(u, v) \in \mathcal{SU} - \{(i, j)\}$, are all above the prescribed thresholds SINR^{th}_{PU} and SINR^{th}_{SU}, respectively, i.e.

$$
\begin{aligned}
(E_i, \mathbf{c}_{ij})^{opt} = & \underset{E_i>0,\mathbf{c}_{ij}\in\mathbb{R}^G}{\arg\max} \; E_i \mathbf{c}_{ij}^T \mathbf{A}^{-1}_{\backslash(i,j)} \mathbf{c}_{ij} \qquad (10)\\
\text{s.t. } & E_l L^2_{lk} \mathbf{s}^T_{lk} \mathbf{A}^{-1}_{\backslash(l,k)} \mathbf{s}_{lk} \geq \text{SINR}^{th}_{PU}, \; \forall(l, k) \in \mathcal{PU},\\
& E_u L^2_{uv} \mathbf{c}^T_{uv} \mathbf{A}^{-1}_{\backslash(u,v)} \mathbf{c}_{uv} \geq \text{SINR}^{th}_{SU}, \; \forall(u, v) \in \mathcal{SU} - \{(i, j)\},\\
& \mathbf{c}^T_{ij}\mathbf{c}_{ij} = 1, \; E_i \leq E_{max},
\end{aligned}
$$

where E_{max} denotes the maximum allowable bit energy for the secondary user.

The optimization task of maximizing a quadratic objective function ($\mathbf{A}^{-1}_{\backslash(i,j)}$ is positive definite) subject to the constraints in (10) is, unfortunately, a nonconvex NP-hard optimization problem [29]. In the following, we delve into the details of the problem and develop a realizable solution.

E. Cognitive Secondary Channel Design

Using the matrix inversion lemma on $\mathbf{A}^{-1}_{\backslash(l,k)}$ and $\mathbf{A}^{-1}_{\backslash(u,v)}$, respectively, we can express the key quadratic constraint expressions in (10), respectively, as

$$\mathbf{s}_{lk}^T \mathbf{A}^{-1}_{\backslash(l,k)} \mathbf{s}_{lk} = \frac{\mathbf{s}_{lk}^T \mathbf{A}_{lk}^{-1} \mathbf{s}_{lk}}{1 - E_l L_{lk}^2 \mathbf{s}_{lk}^T \mathbf{A}_{lk}^{-1} \mathbf{s}_{lk}}, \forall (l,k) \in \mathcal{PU}, \tag{11}$$

$$\mathbf{c}_{uv}^T \mathbf{A}^{-1}_{\backslash(u,v)} \mathbf{c}_{uv} = \frac{\mathbf{c}_{uv}^T \mathbf{A}_{uv}^{-1} \mathbf{c}_{uv}}{1 - E_u L_{uv}^2 \mathbf{c}_{uv}^T \mathbf{A}_{uv}^{-1} \mathbf{c}_{uv}},$$
$$\forall (u,v) \in \mathcal{SU} - \{(i,j)\}, \tag{12}$$

where we recall that $\mathbf{A}_{lk} = \mathbb{E}\{\mathbf{y}_{lk}^{(p)} \mathbf{y}_{lk}^{(p)T}\}$ and $\mathbf{A}_{uv} = \mathbb{E}\{\mathbf{y}_{uv}^{(s)} \mathbf{y}_{uv}^{(s)T}\}$. Then, the primary SINR constraints in (10) become

$$\mathbf{s}_{lk}^T \mathbf{A}_{lk}^{-1} \mathbf{s}_{lk} \geq \frac{\mathrm{SINR}_{PU}^{th}}{E_l L_{lk}^2 + \mathrm{SINR}_{PU}^{th} E_l L_{lk}^2} \triangleq \alpha_{lk},\ \forall (l,k) \in \mathcal{PU}, \tag{13}$$

while the secondary SINR constraints are given by

$$\mathbf{c}_{uv}^T \mathbf{A}_{uv}^{-1} \mathbf{c}_{uv} \geq \frac{\mathrm{SINR}_{SU}^{th}}{E_u L_{uv}^2 + \mathrm{SINR}_{SU}^{th} E_u L_{uv}^2} \triangleq \gamma_{uv},$$
$$\forall (u,v) \in \mathcal{SU} - \{(i,j)\}. \tag{14}$$

The optimization problem (10) can be rewritten as

$$(E_i, \mathbf{c}_{ij})^{opt} = \underset{E_i > 0, \mathbf{c}_{ij} \in \mathbb{R}^G}{\arg\max} \ E_i \mathbf{c}_{ij}^T \mathbf{A}^{-1}_{\backslash(i,j)} \mathbf{c}_{ij} \tag{15}$$
$$\begin{aligned} \text{s.t.} \quad & \mathbf{s}_{lk}^T \mathbf{A}_{lk}^{-1} \mathbf{s}_{lk} \geq \alpha_{lk}, \quad \forall (l,k) \in \mathcal{PU}, \\ & \mathbf{c}_{uv}^T \mathbf{A}_{uv}^{-1} \mathbf{c}_{uv} \geq \gamma_{uv}, \quad \forall (u,v) \in \mathcal{SU} - \{(i,j)\}, \\ & \mathbf{c}_{ij}^T \mathbf{c}_{ij} = 1,\ E_i \leq E_{max}. \end{aligned}$$

Using the matrix inversion lemma on $\mathbf{A}_{lk}^{-1}$ and $\mathbf{A}_{uv}^{-1}$ this time, we can express the optimization constraints as explicit functions of signature vector of the secondary link $\mathbf{c}_{ij}$, i.e.

$$\mathbf{s}_{lk}^T \mathbf{A}_{lk}^{-1} \mathbf{s}_{lk} \geq \frac{E_i L_{ik}^2 \mathbf{s}_{lk} \mathbf{A}^{-1}_{lk\backslash(i,j)} \mathbf{c}_{ij} \mathbf{c}_{ij}^T \mathbf{A}^{-1}_{lk\backslash(i,j)} \mathbf{s}_{lk}}{1 + E_i L_{ik}^2 \mathbf{c}_{ij}^T \mathbf{A}^{-1}_{lk\backslash(i,j)} \mathbf{c}_{ij}} + \alpha_{lk},$$
$$\forall (l,k) \in \mathcal{PU}, \tag{16}$$

$$\mathbf{c}_{uv}^T \mathbf{A}_{uv}^{-1} \mathbf{c}_{uv} \geq \frac{E_i L_{iv}^2 \mathbf{c}_{uv} \mathbf{A}^{-1}_{uv\backslash(i,j)} \mathbf{c}_{ij} \mathbf{c}_{ij}^T \mathbf{A}^{-1}_{uv\backslash(i,j)} \mathbf{c}_{uv}}{1 + E_i L_{ik}^2 \mathbf{c}_{ij}^T \mathbf{A}^{-1}_{uv\backslash(i,j)} \mathbf{c}_{ij}} + \gamma_{uv},$$
$$\forall (u,v) \in \mathcal{SU} - \{(i,j)\}. \tag{17}$$

For notational simplicity, define the $G \times G$ matrices

$$\mathbf{B}_{lk} \triangleq L_{ik}^2 \mathbf{A}_{lk\backslash(i,j)}^{-1} \mathbf{s}_{lk} \mathbf{s}_{lk}^T \mathbf{A}_{lk\backslash(i,j)}^{-1} - \beta_{lk} L_{ik}^2 \mathbf{A}_{lk\backslash(i,j)}^{-1} \tag{18}$$

$$\mathbf{B}_{uv} \triangleq L_{iv}^2 \mathbf{A}_{uv\backslash(i,j)}^{-1} \mathbf{c}_{uv} \mathbf{c}_{uv}^T \mathbf{A}_{uv\backslash(i,j)}^{-1} - \beta_{uv} L_{iv}^2 \mathbf{A}_{uv\backslash(i,j)}^{-1} \tag{19}$$

where

$$\beta_{lk} \triangleq \mathbf{s}_{lk}^T \mathbf{A}_{lk\backslash(i,j)}^{-1} \mathbf{s}_{lk} - \alpha_{lk}, \quad \forall (l,k) \in \mathcal{PU}, \tag{20}$$

$$\beta_{uv} \triangleq \mathbf{c}_{uv}^T \mathbf{A}_{uv\backslash(i,j)}^{-1} \mathbf{c}_{uv} - \gamma_{uv}, \quad \forall (u,v) \in \mathcal{SU} - \{(i,j)\}. \tag{21}$$

Then, the optimization problem in (15) can be rewritten -for one more time- as

$$\begin{aligned} \mathbf{x}^{opt} &= \arg\max_{\mathbf{x} \in \mathbb{R}^G} \ \mathbf{x}^T \mathbf{A}_{\backslash(i,j)}^{-1} \mathbf{x} \\ \text{s.t.} \quad & \mathbf{x}^T \mathbf{B}_{lk} \mathbf{x} - \beta_{lk} \le 0, \forall (l,k) \in \mathcal{PU}, \\ & \mathbf{x}^T \mathbf{B}_{uv} \mathbf{x} - \beta_{uv} \le 0, \forall (u,v) \in \mathcal{SU} - \{(i,j)\}, \\ & \mathbf{x}^T \mathbf{x} \le E_{max} \end{aligned} \tag{22}$$

where $\mathbf{x}$ is the amplitude-including signature vector of secondary link (i,j), $\mathbf{x} \triangleq \sqrt{E_i}\mathbf{c}_{ij}$. From the perspective of computational effort, however, (i) $\mathbf{B}_{lk}$ and $\mathbf{B}_{uv}$ are not necessarily positive semidefinite, hence the problem in (22) is in general a non-convex quadratically constrained quadratic program (non-convex QCQP), and (ii) the complexity of a solver of (22) is exponential in the dimension G (NP-hard problem).

Using the commutative property of trace, we are able to represent the optimization problem in (22) as

$$\begin{aligned} \mathbf{X}^{opt} &= \arg\max_{\mathbf{X} \in \mathbb{R}^{G \times G}} Tr\{\mathbf{A}_{\backslash(i,j)}^{-1} \mathbf{X}\} \\ \text{s.t.} \quad & Tr\{\mathbf{B}_{lk}\mathbf{X}\} \le \beta_{lk}, \quad \forall (l,k) \in \mathcal{PU}, \\ & Tr\{\mathbf{B}_{uv}\mathbf{X}\} \le \beta_{uv}, \quad \forall (u,v) \in \mathcal{SU} - \{(i,j)\}, \\ & Tr\{\mathbf{X}\} \le E_{max}, \ \mathbf{X} \succeq \mathbf{0}, \ rank(\mathbf{X}) = 1. \end{aligned} \tag{23}$$

where $\mathbf{X} \triangleq \mathbf{x}\mathbf{x}^T$ and $\mathbf{X} \succeq \mathbf{0}$ denotes that the matrix $\mathbf{X}$ is positive semidefinite.

To effectively attack the problem anyway, the problem in (23) is relaxed to a semidefinite program by dropping the rank constraint, i.e.

$$\begin{aligned} \mathbf{X}' &= \arg\max_{\mathbf{X} \in \mathbb{R}^{G \times G}} Tr\{\mathbf{A}_{\backslash(i,j)}^{-1} \mathbf{X}\} \\ \text{s.t.} \quad & Tr\{\mathbf{B}_{lk}\mathbf{X}\} \le \beta_{lk}, \quad \forall (l,k) \in \mathcal{PU}, \\ & Tr\{\mathbf{B}_{uv}\mathbf{X}\} \le \beta_{uv}, \quad \forall (u,v) \in \mathcal{SU} - \{(i,j)\}, \\ & Tr\{\mathbf{X}\} \le E_{max}, \ \mathbf{X} \succeq \mathbf{0}. \end{aligned} \tag{24}$$

Then, (24) is a convex problem that can be solved using semidefinite programming within the error ϵ in $O(G^4 \log 1/\epsilon)$ time. The solution $\mathbf{X}''$ returned by semidefinite programming makes the objective function $Tr\{\mathbf{A}_{\setminus(i,j)}^{-1}\mathbf{X}\}$ attain a value within $(Tr\{\mathbf{A}_{\setminus(i,j)}^{-1}\mathbf{X}'\} - \epsilon, Tr\{\mathbf{A}_{\setminus(i,j)}^{-1}\mathbf{X}'\})$. Of course, because of the constraint relaxation itself the objective function evaluated at the optimum point $\mathbf{X}'$ in (24) is just an upper bound on the value of the objective function evaluated at the optimum point of interest $\mathbf{X}^{opt}$ of (23), $Tr\{\mathbf{A}_{\setminus(i,j)}^{-1}\mathbf{X}^{opt}\} \leq Tr\{\mathbf{A}_{\setminus(i,j)}^{-1}\mathbf{X}'\}$. Since $\mathbf{X}'$ is not available, we have $\mathbf{X}''$ with $Tr\{\mathbf{A}_{\setminus(i,j)}^{-1}\mathbf{X}''\} \in (Tr\{\mathbf{A}_{\setminus(i,j)}^{-1}\mathbf{X}'\} - \epsilon, Tr\{\mathbf{A}_{\setminus(i,j)}^{-1}\mathbf{X}'\})$. So far, for the cognitive design of a code-division secondary link, first, for the given primary and secondary SINR-QoS thresholds $\text{SINR}_{PU}^{th} > 0$ and $\text{SINR}_{SU}^{th} > 0$, we test whether β_{lk}, $\forall(l,k) \in \mathcal{PU}$ and $\beta_{uv}, \forall(u,v) \in \mathcal{SU} - \{(i,j)\}$ are all greater than zero. If this is not true, then the SINR-QoS constraints cannot be satisfied and outright no secondary transmission is allowed. Otherwise, we solve the problem (24) by semidefinite programming with the return $\mathbf{X}''$. If the rank of $\mathbf{X}''$ is 1 with eigenvalue, eigenvector pair $(\lambda_1, \mathbf{v}_1)$, then we already have our secondary link design with signature $\mathbf{c}_{ij} = \mathbf{v}_1$ and transmission bit energy $E_i = \lambda_1$. If the rank of $\mathbf{X}''$ is not 1, further work is needed as described below.

When $\mathbf{X}'$ of (24) (or in practice $\mathbf{X}''$) is of rank 2 or more, we can not find the direct mapping from $\mathbf{X}'$ of (24) to $\mathbf{x}^{opt}$ in (22). Under this case, we may switch the search for an optimal vector in (22) to a search for an optimal probability density function (pdf) of vectors that maximizes the average objective function subject to average constraints, i.e.

$$\begin{aligned} f^{opt}(\mathbf{x}) = \max_{f(\mathbf{x})} \ & \mathbb{E}\{\mathbf{x}^T \mathbf{A}_{\setminus(i,j)}^{-1} \mathbf{x}\} \\ \text{s.t.} \quad & \mathbb{E}\{\mathbf{x}^T \mathbf{B}_{lk} \mathbf{x}\} \leq \beta_{lk}, \forall(l,k) \in \mathcal{PU}, \\ & \mathbb{E}\{\mathbf{x}^T \mathbf{B}_{uv} \mathbf{x}\} \leq \beta_{uv}, \forall(u,v) \in \mathcal{SU} - \{(i,j)\}, \\ & \mathbb{E}\{\mathbf{x}^T \mathbf{x}\} \leq E_{max} \end{aligned} \tag{25}$$

where $f(\mathbf{x})$ denotes the probability density function of $\mathbf{x}$ and $\mathbb{E}\{\cdot\}$ denotes statistical expectation. Using the commutative property between trace and expectation operators, the pdf optimization problem in (25) takes the equivalent form

$$\begin{aligned} f^{opt}(\mathbf{x}) = \arg\max_{f(\mathbf{x})} \ & Tr\{\mathbf{A}_{\setminus(i,j)}^{-1} \mathbb{E}\{\mathbf{x}\mathbf{x}^T\}\} \\ \text{s.t.} \quad & \mathbf{B}_{lk}\mathbb{E}\{\mathbf{x}\mathbf{x}^T\} \leq \beta_{lk}, \forall(l,k) \in \mathcal{PU}, \\ & \mathbf{B}_{uv}\mathbb{E}\{\mathbf{x}\mathbf{x}^T\} \leq \beta_{uv}, \forall(u,v) \in \mathcal{SU} - \{(i,j)\}, \\ & \mathbb{E}\{\mathbf{x}\mathbf{x}^T\} \leq E_{max}. \end{aligned} \tag{26}$$

We can show that $f^{opt}(\mathbf{x})$ is in fact Gaussian with $\mathbf{0}$ mean and covariance matrix $\mathbf{X}'$, $f^{opt}(\mathbf{x}) = \mathcal{N}(\mathbf{0}, \mathbf{X}')$. With $\mathbf{X}''$ as a close approximation of $\mathbf{X}'$, we can draw now a sequence of samples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_P$ from $\mathcal{N}(\mathbf{0}, \mathbf{X}'')$. We test all of them for "feasibility" on whether all constraints of (22) are satisfied and among the feasible vectors (if any) we choose the one, say $\mathbf{x}^{(0)}$, with maximum $\mathbf{x}^T\mathbf{A}_{\setminus(i,j)}^{-1}\mathbf{x}$ objective function value. We could have suggested at this time a cognitive secondary link design with $\sqrt{E_i}\mathbf{c}_{ij} = \mathbf{x}^{(0)}$. Instead, we will use $\mathbf{x}^{(0)}$ as an initialization point to an iterative procedure below that will lead to a improved link design vector. First we express $\mathbf{A}_{\setminus(i,j)}$ as

$$\mathbf{A}_{\setminus(i,j)} = \mathbf{S}\Sigma\,\mathbf{S}^T + \sigma^2\mathbf{I} \tag{27}$$

where $\mathbf{S} \stackrel{\triangle}{=} [\{\mathbf{s}_{lk}, \forall(l,k) \in \mathcal{PU}\}, \{\mathbf{c}_{uv}, \forall(u,v) \in \mathcal{SU} - \{(i,j)\}]$ denotes the matrix with columns the signatures of the primary users and ongoing secondary users; $\Sigma \stackrel{\triangle}{=} \mathrm{diag}(\{E_l L_{lj}^2, \forall(l,k) \in \mathcal{PU}\}, \{E_u L_{uj}^2, \forall(u,v) \in \mathcal{SU} - \{(i,j)\})$. Using the matrix inversion lemma,

$$\mathbf{A}_{\setminus(i,j)}^{-1} = \frac{1}{\sigma^2}\mathbf{I} - \frac{1}{\sigma^4}\mathbf{S}(\Sigma^{-1} + \frac{1}{\sigma^2}\mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T. \tag{28}$$

Substitution of (28) in the objective function of (22) leads to

$$\mathbf{x}^T\mathbf{A}_{\setminus(i,j)}^{-1}\mathbf{x} = \frac{1}{\sigma^2}\mathbf{x}^T\mathbf{x} - \frac{1}{\sigma^4}\mathbf{x}^T\mathbf{Q}\,\mathbf{x} \tag{29}$$

where $\mathbf{Q} \stackrel{\triangle}{=} \mathbf{S}(\Sigma^{-1} + \frac{1}{\sigma^2}\mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T$. In (29), the first term $\frac{1}{\sigma^2}\mathbf{x}^T\mathbf{x}$ is a convex function while the second term $-\frac{1}{\sigma^4}\mathbf{x}^T\mathbf{Q}\,\mathbf{x}$ is a concave function (which implies that $\frac{1}{\sigma^4}\mathbf{x}^T\mathbf{Q}\,\mathbf{x}$ is convex). Based on the first-order condition of convex functions, we have

$$\mathbf{x}^T\mathbf{x} \geq 2\mathbf{x}^{(0)T}\mathbf{x} - \mathbf{x}^{(0)T}\mathbf{x}^{(0)} \tag{30}$$

where $\mathbf{x}^{(0)}$ denotes an initial feasible vector. Then, we combine (29) and (30) and form an optimization problem that maximizes the following concave function

$$\frac{2}{\sigma^2}\mathbf{x}^{(0)T}\mathbf{x} - \frac{1}{\sigma^4}\mathbf{x}^T\mathbf{Q}\,\mathbf{x} - \frac{1}{\sigma^2}\mathbf{x}^{(0)T}\mathbf{x}^{(0)} \tag{31}$$

that leads to a *suboptimum* solution for our original problem in (22). To maximize (31) in view of our constraints in (22), we restrict all non-convex constraints into convex sets (linearization). In particular, we consider the non-convex constraints

$$\mathbf{x}^T\mathbf{B}_{lk}\,\mathbf{x} \leq \beta_{lk}, \quad \forall(l,k) \in \mathcal{I}_{nc}^{(p)}, \tag{32}$$

$$\mathbf{x}^T\mathbf{B}_{uv}\,\mathbf{x} \leq \beta_{uv}, \quad \forall(u,v) \in \mathcal{I}_{nc}^{(s)}, \tag{33}$$

where $\mathcal{I}_{nc}^{(p)}$ and $\mathcal{I}_{nc}^{(s)}$ denote sets of all pairs $(l,k) \in \mathcal{PU}$ and $(u,v) \in \mathcal{SU} - \{(i,j)\}$ for which the quadratic constraint is non-convex function, respectively. Then, we decompose the matrices $\mathbf{B}_{lk}$ and $\mathbf{B}_{uv}$ into its positive and negative parts that are positive semidefinite, i.e. $\mathbf{B}_{lk} = \mathbf{B}_{lk}^+ - \mathbf{B}_{lk}^-$, where $\mathbf{B}_{lk}^+ = L_{lj}^2\mathbf{A}_{\setminus(i,j)}^{-1}\mathbf{s}_{lk}\mathbf{s}_{lk}^T\mathbf{A}_{\setminus(i,j)}^{-1}$, $\mathbf{B}_{lk}^- = \beta_{lk}L_{lj}^2\mathbf{A}_{\setminus(i,j)}^{-1}$; $\mathbf{B}_{uv} = \mathbf{B}_{uv}^+ - \mathbf{B}_{uv}^-$, where $\mathbf{B}_{uv}^+ = L_{uj}^2\mathbf{A}_{\setminus(i,j)}^{-1}\mathbf{c}_{uv}\mathbf{c}_{uv}^T\mathbf{A}_{\setminus(i,j)}^{-1}$, $\mathbf{B}_{uv}^- = \beta_{uv}L_{uj}^2\mathbf{A}_{\setminus(i,j)}^{-1}$. Therefore, the original non-convex constraints can be written as

$$\mathbf{x}^T\mathbf{B}_{lk}^+\,\mathbf{x} - \beta_{lk} \leq \mathbf{x}^T\mathbf{B}_{lk}^-\,\mathbf{x}, \quad \forall(l,k) \in \mathcal{I}_{nc}^{(p)}, \tag{34}$$

$$\mathbf{x}^T \mathbf{B}_{uv}^+ \mathbf{x} - \beta_{uv} \leq \mathbf{x}^T \mathbf{B}_{uv}^- \mathbf{x}, \quad \forall (u,v) \in \mathcal{I}_{nc}^{(s)}, \tag{35}$$

where both sides of the inequalities are convex quadratic functions. Linearization of the right-hand side of (34) around the vector $\mathbf{x}^{(0)}$ leads to

$$\mathbf{x}^T \mathbf{B}_{lk}^+ \mathbf{x} - \beta_{lk} \leq \mathbf{x}^{(0)T} \mathbf{B}_{lk}^- \mathbf{x}^{(0)} + 2\mathbf{x}^{(0)T} \mathbf{B}_{lk}^- (\mathbf{x} - \mathbf{x}^{(0)}), \\ \forall (l,k) \in \mathcal{I}_{nc}^{(p)}, \tag{36}$$

$$\mathbf{x}^T \mathbf{B}_{uv}^+ \mathbf{x} - \beta_{uv} \leq \mathbf{x}^{(0)T} \mathbf{B}_{uv}^- \mathbf{x}^{(0)} + 2\mathbf{x}^{(0)T} \mathbf{B}_{uv}^- (\mathbf{x} - \mathbf{x}^{(0)}), \\ \forall (u,v) \in \mathcal{I}_{nc}^{(s)}. \tag{37}$$

In (36) and (37), the right-hand side is an affine lower bound on the original function. It is thus implied that the resulting constraints are convex and more conservative than the original ones, hence the feasible set of the linearized problem is a convex subset of the original feasible set. Thus, by linearizing the concave parts of all constraints, we obtain a set of convex constraints that are tighter than the original non-convex ones. Now, the original optimization problem takes the form

$$\begin{aligned} \mathbf{x}^{(1)} = \arg \max_{\mathbf{x} \in \mathbb{R}^L} \; & \frac{2}{\sigma^2}\mathbf{x}^{(0)T}\mathbf{x} - \frac{1}{\sigma^4}\mathbf{x}^T \mathbf{Q} \mathbf{x} - \frac{1}{\sigma^2}\mathbf{x}^{(0)T}\mathbf{x}^{(0)} \\ \text{s.t. } & \mathbf{x}^T \mathbf{B}_{lk}^+ \mathbf{x} - \mathbf{x}^{(0)T} \mathbf{B}_{lk}^- (2\mathbf{x} - \mathbf{x}^{(0)}) \leq \beta_{lk}, \; \forall (l,k) \in \mathcal{I}_{nc}^{(p)}, \\ & \mathbf{x}^T \mathbf{B}_{lk} \mathbf{x} \leq \beta_{lk}, \quad \forall (l,k) \in \mathcal{PU} - \mathcal{I}_{nc}^{(p)}, \\ & \mathbf{x}^T \mathbf{B}_{uv}^+ \mathbf{x} - \mathbf{x}^{(0)T} \mathbf{B}_{uv}^- (2\mathbf{x} - \mathbf{x}^{(0)}) \leq \beta_{uv}, \; \forall (u,v) \in \mathcal{I}_{nc}^{(s)}, \\ & \mathbf{x}^T \mathbf{B}_{uv} \mathbf{x} \leq \beta_{uv}, \quad \forall (u,v) \in \mathcal{SU} - \mathcal{I}_{nc}^{(s)} - \{(i,j)\}, \\ & \mathbf{x}^T \mathbf{x} \leq E_{max}. \end{aligned} \tag{38}$$

The problem in (38) is a convex QCQP problem and can be solved efficiently by standard convex system solvers [30] to produce a new feasible vector $\mathbf{x}^{(1)}$. The objective function $\mathbf{x}^T \mathbf{A}_{\backslash(i,j)}^{-1} \mathbf{x}$ in (22) evaluated at $\mathbf{x}^{(1)}$ takes a value that is larger than or equal to its value at $\mathbf{x}^{(0)}$. Repeating iteratively the linearization procedure, we can obtain a sequence of feasible vectors $\mathbf{x}^{(0)}$, $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}$ with non-decreasing values of the objective function in (22). As demonstrated by experimental results in [31], it is observed that eight or nine iterations are enough for effective convergence. After numerical convergence, the secondary link (i,j) is suggested with signature $\mathbf{c}_{ij} = \mathbf{x}^{(N)}/\|\mathbf{x}^{(N)}\|$ and bit energy $E_i = \|\mathbf{x}^{(N)}\|^2$. If the secondary transmission bit energy and signature vector returned by our algorithm satisfy the constraint $E_i L_{i,j}^2 \mathbf{c}_{ij}^T \mathbf{A}_{\backslash(i,j)}^{-1} \mathbf{c}_{ij} \geq \text{SINR}_{SU}^{th}$, the secondary link (i,j) with E_i and $\mathbf{c}_{ij}$ is considered as candidate for transmission. Otherwise, transmission over the path (i,j) is not allowed.

Our proposed scheme is summarized in Algorithm 2.

The complexity of the optimization in the physical layer (i.e. optimizing the secondary transmission bit energy and signature vector) is dominated by the complexity of solving the semidefinite program in (24). The computation

Algorithm 2 Cognitive Code-division Channelization.

if $\beta_{lk} \geq 0, \forall (l,k) \in \mathcal{PU}$ and $\beta_{uv} \geq 0, \forall (u,v) \in \mathcal{SU} - \{(i,j)\}$ **then**
  Run SDP optimizer
  if Rank($\mathbf{X}''$)=1 **then**
    Obtain $(\lambda_1, \mathbf{v}_1)$ from eigen decomposition of $\mathbf{X}''$
    Assign $E_i \leftarrow \lambda_1$ and $\mathbf{c}_{ij} \leftarrow \mathbf{v}_1$
  else
    Draw $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_P$ from $\mathcal{N}(\mathbf{0}, \mathbf{X}'')$
    if Any feasible sample $\mathbf{x}_p$, $p \in \{1, \ldots, P\}$ that satisfies $\mathbf{x}_p^T \mathbf{B}_{lk} \mathbf{x}_p \;\; \forall\, (l,k) \in \mathcal{PU}$ & $\mathbf{x}_p^T \mathbf{B}_{uv} \mathbf{x}_p \;\; \forall\, (u,v) \in \mathcal{SU} - \{(i,j)\}$ **then**
      $\mathbf{x}^{(0)} \leftarrow$ feasible $\mathbf{x}_p$ with maximum $\mathbf{x}_p^T \mathbf{A}_{\setminus(i,j)}^{-1} \mathbf{x}_p$ value
      Run iteratively linearized optimizer with convergence point $\mathbf{x}^{(N)}$
      Assign $E_i \leftarrow \|\mathbf{x}^{(N)}\|^2$ and $\mathbf{c}_{ij} \leftarrow \mathbf{x}^{(N)} / \|\mathbf{x}^{(N)}\|$
    else
      No feasible solution by assigning $E_i \leftarrow 0$ and $\mathbf{c}_{ij} \leftarrow \mathbf{0}$
      Return
    end if
  end if
  if $E_i L_{i,j}^2 \mathbf{c}_{ij}^T \mathbf{A}_{\setminus(i,j)}^{-1} \mathbf{c}_{ij} \geq SINR_{SU}^{th}$ **then**
    Output the solution E_i and $\mathbf{c}_{ij}$
  else
    No feasible solution by assigning $E_i \leftarrow 0$ and $\mathbf{c}_{ij} \leftarrow \mathbf{0}$
  end if
else
  No feasible solution by assigning $E_i \leftarrow 0$ and $\mathbf{c}_{ij} \leftarrow \mathbf{0}$
end if
Return the link design $(E_i, \mathbf{c}_{ij})$ as candidate for transmission.

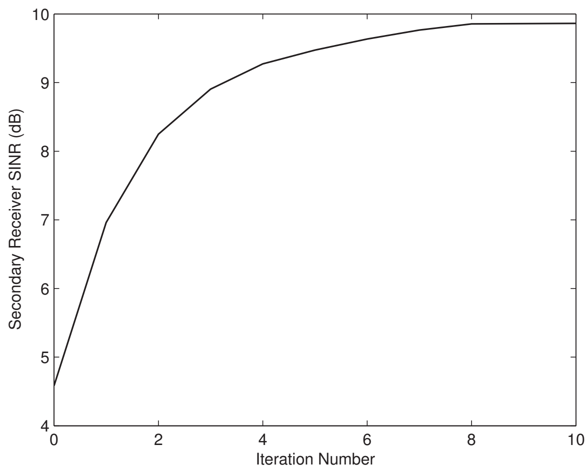


Fig. 2: Secondary receiver SINR as a function of the iteration step of the linearized optimizer initialized at the best feasible sample out of $P = 20$ drawings from the $\mathcal{N}(\mathbf{0}, \mathbf{X}'')$ pdf.

complexity of the algorithm for the physical layer is $\mathcal{O}(G^4 \log 1/\epsilon)$. [31] gives more detailed complexity analysis on the optimization in the physical layer. For any of the K feasible next hops, each node executes ROCH, which clearly has polynomial complexity in the number of $\mathcal{O}(G^4 \log 1/\epsilon)$. Conversely, the centralized algorithm of this family of schedule and routing has worst-case exponential complexity [28].

Here, we demonstrate the effective convergence of proposed algorithm through simulation studies. In the simulation, the transmission SNRs of primary links are all set equal to 15 dB. The number of active primary users is set to 16 (fully loaded). All the signatures for primary links are generated from a minimum total-squared correlation optimal binary signature set which achieves the Karystinos-Pados (KP) bound [32]-[34]. The SINR threshold is set to 3 dB for both primary and secondary users. In Fig. 2, we plot the secondary receiver *average* SINR for the experimental instants of $rank(\mathbf{X}'') > 1$ as a function of the iteration of the optimizer initialized at the point/design $\mathbf{x}^{(0)}$ that is the best out of $P = 20$ samples drawn from the $\mathcal{N}(\mathbf{0}, \mathbf{X}'')$ pdf. It is pleasing to observe that eight or nine iterations are enough for effective convergence and the improvement is more than 5 dB.

IV. Results and Discussion

In this section, we evaluate the performance of ROCH in a multi-hop cognitive radio network. We have developed an object-oriented packet-level discrete-event simulator, which models in detail all layers of the communication protocol stack, including routing, medium access control and spread-spectrum channelization as described in this project.

A. Performance of Secondary Dynamic Spectrum Access

We consider K active links (including primary and secondary links) with signature length (system processing gain) $G = 16$. We are interested in establishing a secondary code-division transmitter/receiver pair when K varies from 14 to 20. All link signatures are generated from a minimum total-squared-correlation optimal binary signature set which achieves the Karystinos-Pados (KP) bound for each (K, G) pair of values[3]. The transmission SNRs of K active links are all set equal to $\frac{E_i}{\sigma^2} = 15\,dB$, $i = 1, 2, \ldots K$; the maximum allowable transmission SNR for the secondary link is set to $\frac{E_{max}}{\sigma^2} = 15\,dB$. The channels are modeled as Rayleigh fading. The channel coefficients are taken to be the magnitude of independent complex Gaussian random variables with mean 0 and variance 1. The receiver SINR thresholds for primary and secondary links are set to $\text{SINR}_{PU}^{th} = 3\,dB$ and $\text{SINR}_{SU}^{th} = 3\,dB$, respectively. When random vector drawing is necessary, $P = 20$ test vector points are generated.

In Fig. 3(a), we plot, as a function of the number of active links K, the percentage of times that one more secondary link is enabled directly under the proposed algorithm in Section V as well as the random code assignment (RCA) scheme where the candidate signature is randomly generated with unit norm and transmission bit energy is set to be the maximum allowable value E_{max}. We observe that our proposed algorithm in Section VI offers more opportunities for cognitive secondary transmission than RCA.

In Fig. 3(b), to gain visual insight into operation of the network we plot the *instantaneous* receiver SINR of a primary active link and the candidate secondary link for the case $K = 17$ over an experimental data record sequence of 1000 Rayleigh fading channel realizations. Missing secondary signal SINR values indicate instances when no feasible solution was returned. The proposed scheme highly increases the probability of secondary transmission compared to RCA. When secondary transmissions do occur for both schemes, joint power and sequence optimization executed by the proposed scheme results in superior SINR performance for the secondary receiver over RCA.

B. Network Performance Evaluation

A grid topology of 49 secondary nodes and 14 active primary links is deployed in a 5000 m $\times$5000 m area. The spreading code length is set to 16 for both primary

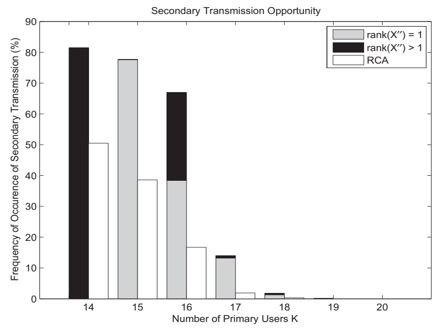
[3]For $G = 16$, when $K \leq G$ the KP-optimal sequences coincide with the familiar Walsh-Hadamard signature codes.

and secondary users. Active primary links use pre-assigned code sequences as described in the previous section. We initiate CBR traffic sessions between randomly selected but disjoint source-destination pairs among the 49 nodes. Parameters θ_1 and θ_2 in (9) are set to 5 and 10, respectively. Rayleigh fading channel is used and the path loss exponent is set to four. We compare the performance of ROCH with three alternative schemes. All alternative schemes rely on the same knowledge of the environment as ROCH. In particular, we consider the solution SP-SIG where routing is based on the shortest path with dynamic signature and power allocation (as proposed in Section VI). The other two solutions use RCA with fixed power allocation. We consider BP-RCA as the solution where routing is based on the same utility function as ROCH but with random code assignment, and SP-RCA as the solution where routing is based on the shortest path with random code assignment.

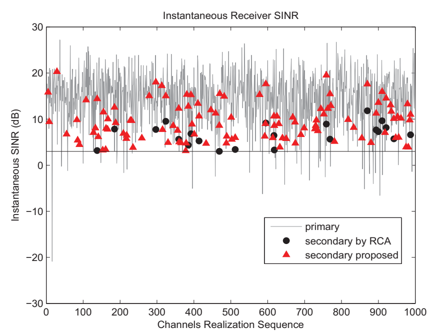
We compare the four solutions by varying the number of sessions injected into the network and plot the network throughput (sum of individual session throughputs). Figures 3(c) and 4(a) show the impact of the number of sessions injected into the network on the network throughput. The traffic load per session is set to 2 Mbit/s and 4 Mbit/s. As shown in both figures, ROCH achieves the highest throughput. The improvement obtained by ROCH is more visible when the number of active sessions increases. Solutions with adaptive signature design, i.e., the proposed solution ROCH and SP-SIG, outperform RCA-based solutions, i.e., BP-RCA and SP-RCA. With the same signature optimization algorithm, ROCH outperforms SP-SIG since SP-SIG restricts packets forwarding to the receiver that is the closest to the destination, even if the link capacity is very low or the receiver is heavily congested.

Fig. 4(b) shows the impact of source data rate per session on the performance of throughput. We evaluate the throughput performance as the traffic load per session increases from 1 Mbit/s to 4 Mbit/s. As shown in Fig. 4(b), ROCH obtains a significant throughput gain.

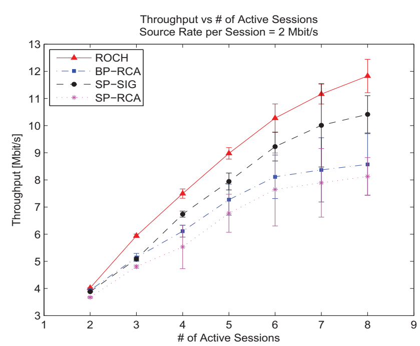
Fig. 4(c) shows the delay performance for the four solutions with traffic load 2 Mbit/s per session. In general, solutions with adaptive signature design (ROCH and SP-SIG) outperforms RCA-based solutions (BP-RCA and SP-RCA) in terms of delay, and the delay performance gap between the two grows as the number of sessions increases.



(a)

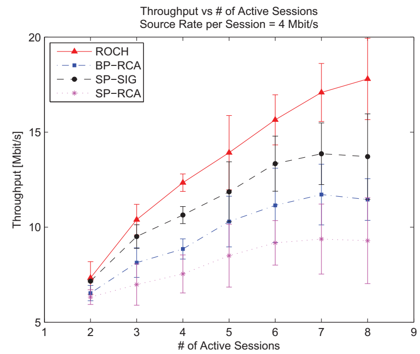


(b)

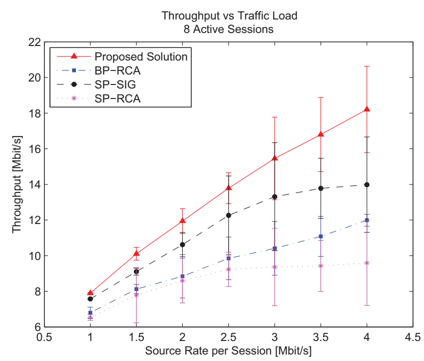


(c)

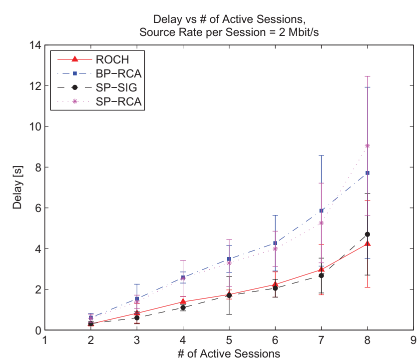
Fig. 3: (a) Secondary transmission percentage as a function of the number of active links under Cases $rank(\mathbf{X}'') = 1$ and > 1 (the study includes also the random code assignment scheme); (b) Instantaneous output SINR of a primary signal against primary SINR-QoS threshold SINR_{PU}^{th} (thick line) and instantaneous output SINR of secondary signal; (c) Throughput vs. Number Active Sessions.



(a)



(b)



(c)

Fig. 4: (a) Throughput vs. Number Active Sessions; (b) Throughput vs. Traffic Load; (c) Delay vs. Number Active Sessions.

C. RcUBe: Real-Time Reconfigurable Radio Framework with Self-Optimization Capabilities

As of today, existing wireless networking standards are inherently *hardware-based*. The majority of deployed networks are neither capable of keeping up with the quickly evolving pace of wireless standards, nor of adopting application-specific protocol designs, mainly due to their limited programmability and reconfigurability [35, 36]. As a consequence, practical deployment of new protocol amendments or of new standards and their application is prohibitive in terms of both cost and time. There is a profound and urgent need for reconfigurable wireless architectures to overcome the aforementioned constraints and offer the required flexibility.

Moreover, while in recent years there have been significant algorithmic developments in cross-layer network adaptation and resource allocation [37, 38, 39, 40], in most cases existing architectures are not flexible enough to incorporate these advancements other than through ad hoc tweaks. As a consequence, algorithmic developments and architectural and protocol innovation follow parallel paths, with little or no cross-fertilization between the two communities.

For example, among the emerging wireless technologies, cognitive radio (CR) has been envisioned as a technology with a strong potential to increase the spectrum utilization efficiency [41]. Cognitive radio technology relies on the ability of the radio terminal to reconfigure crucial communication parameters (e.g., modulation, carrier frequency, transmission power, among others) to achieve dynamic spectrum access. In recent work [42, 43, 44, 45, 46, 47, 48, 49], the importance and need for reconfigurability between different layers of the protocol stack (cross-layer approach) has been highlighted and demonstrated. However, it is unclear how these developments can be incorporated in existing and future network architectures. Moreover, as mentioned in [50], application-specific (e.g., QoS-featured applications and white-space networking) and context-specific (e.g., video communications) protocol designs require the establishment of more flexible, reconfigurable wireless infrastructures. Evidently, reconfigurability is becoming a crucial feature for wireless networking technologies.

Based on these premises, in this project we introduce *R*eal-time *R*econfigurable *R*adio (RcUBe), a novel architectural radio framework that offers self-optimization and real-time reconfiguration and adaptation capabilities at the PHY, MAC, and network layers of the networking protocol stack. RcUBe introduces an extensive and evolving abstraction for implementing a wide range of protocols. For this purpose, RcUBe adopts the idea of decomposing protocols into primitive building blocks [35, 36, 51, 50]. The novelty of RcUBe is based on its extent and capabilities. First, unlike [35, 36, 51, 50] RcUBe offers real-time reconfigurability at three layers (PHY, MAC, routing). Second, it provides a structured methodology to implement cross-layer algorithms and optimizations spanning multiple layers of the protocol stack. Finally, it provides support for the implementation of real-time self-optimization capabilities that are the basis of state-of-the-art CR algorithms.

The design of RcUBe is modular, in the sense that it preserves a layered protocol stack; and is compatible with TCP/IP (although integration with TCP/IP is out of the scope of this project). However, the architecture of the network node is divided into four distinct, separate, but interacting planes, each logically in charge of a different group of functionalities. These planes are referred to as *decision*, *control*, *data*, and *register* planes. The decision plane consists of user-defined decision algorithms that provide the control logic for real-time protocol optimization, adaptation, and reconfiguration. In the control plane, RcUBe stores the required logic for routing and MAC protocol execution as well as data plane management. Data processing takes place in the data plane, while the register plane stores and manages access to system parameters and environmental information. This division has several advantages. *First*, this structure separates the decision plane, which enables real-time optimization, from execution of the protocol stack. Thus, it becomes possible to define and reconfigure the decision logic on-the-fly without influencing the on-going protocol execution logic. The decision plane consists of a decision engine where user-defined algorithms are executed to enable adaptation and optimization of the protocol stack. *Second*, we decouple data-processing functionalities from their control logic. In other words, we decompose the protocol stack into data and control planes. In this way, we obtain the essential modularity and flexibility for reconfiguring the protocol stack in real-time. *Finally*, we define a register plane, which represents a shared, common storage space of the state variables of protocols handling functionalities at different layers of the protocol stack. The register plane provides a liaison between different planes and also stores system parameters and general purpose variables that can be accessed by multiple protocols.

Our design is implemented and tested on the Universal Software Radio Peripheral (URSP) platform [52], a commercial, low-cost RF front-end. This is in contrast to [51] that uses comparatively expensive WARP [53] boards. C++ and Python languages, supported by the GNU Radio framework, are used for PHY layer implementations, while the overall architecture is implemented in Python. Through these design choices, RcUBe offers easier programmability compared to [35, 51, 36]. The proposed framework is natively decoupled from hardware. In its current instantiation, it mainly runs on a host PC, but it can be adapted to run on any multi-core platform along with any available RF front-end; as well as on soft-core processors implemented on FPGAs. RcUBe is based on a software-defined physical layer, so it provides broad reconfigurability at the PHY layer, as it is not strictly limited to the reconfiguration of the capabilities of selected commodity WLAN cards. Instead, it exploits the PHY layer capabilities of the GNU Radio framework and leverages its modular and flexible structure.

To sum up, the major contributions of this project are:

- the design of a real-time reconfigurable radio framework with support for control logic enabling adaptation and self-optimization;
- real-time reconfigurability for PHY, MAC, and routing protocols;
- real-time cross-layer optimization;
- an extensive abstraction for implementing a wide range of protocols;
- illustration of the unique features and the flexible design of RcUBe by implementing a cross-layer architecture that performs on-the-fly reconfigurations based upon decisions taken in real time by a decentralized optimization algorithm.

D. Related Work

Wireless protocol stacks are traditionally implemented on hardware with a static architecture, which offers little or no room for reconfigurability. To overcome this limitation and enable flexibility on traditional wireless infrastructure, early work adopted the idea of using multiple existing modules on commercial 802.11 cards. For example, MultiMAC [54] is based on the idea of switching between pre-defined MAC protocols. However, this offers only limited flexibility, since only a limited number of MAC protocols can be pre-implemented on a card. FreeMAC [55] and TDMA MAC [56] provide reconfigurability to MAC protocol design. Their flexibility is however limited by the coercive wireless-card specifications.

The idea of using dedicated wireless reconfigurable platforms has also emerged to address the lack of reconfigurability of existing hardware. Software-defined radios (SDRs) such as USRP [52] are a good example of such dedicated platforms. USRP works along with the GNU Radio [57] framework, which offers flexibility thanks to its software-based processing and modular design features. However, this flexibility is limited to the PHY layer. SORA [58] offers flexibility along with high performance through a multi-core processing architecture. However, its design imposes difficulties (in terms of software complexity) in reconfiguring the protocol stack. Moreover, platforms like AirBlue [59] and WARP [53] transfer most of the processing functions to Field-Programmable Gate Arrays (FPGA) to improve performance. AirBlue also has a modular architecture that enables easier modifications in MAC design.

Another family of recent proposals are based on the idea of decomposing wireless MAC protocols into core functional blocks that can be composed through a high-level language to enable reconfigurability [35, 51, 36, 50]. This promising approach is also at the core of the design of RcUBe. In [35], the architecture is built on USRP [57, 52]. The key design idea is to split MAC functionalities between the host processing and the hardware (FPGA) processing units. Hence, time-critical MAC functionalities are run on FPGA for better performance and timing control, while other control functionalities are kept in the host PC. Conversely, in [51], all MAC primitive blocks are implemented on hardware. MAC blocks are implemented (in WARP [53]) on FPGA boards and a PowerPc CPU is employed as the main processing unit of the architecture. In [36], the proposed architecture is built on a resource-constrained commodity WLAN card. Unlike previous work, the authors propose the implementation of an abstract execution machine which, based on conditions and events, controls the execution of primitive building blocks. These blocks are created by decomposing basic MAC protocols and define basic MAC functions (i.e., actions). Therefore, MAC protocols can be flexibly defined as extended finite state machines implemented on commodity wireless cards, thus enabling real-time reprogrammability, without modifying the hardware. In a follow-up work [60], extensions are proposed to reduce the run-time reconfiguration delay, and a control logic is proposed to handle (i.e., load, move, and activate) MAC programs. In [50], the authors propose a programmable dataplane architecture that decomposes wireless protocols into separate processing and decision planes. They argue that their design can work on any multicore DSP processor.

At the network layer, OpenFlow [61] has been primarily proposed for testing experimental routing protocols on already deployed networks, particularly in campus networks. OpenFlow, which is inspired by the internal structure of an Ethernet Switch, consists of three main components, i.e., (i) an internal flow table where each flow is matched with its corresponding action, (ii) a secure channel connecting a switch with an outer controller, and (iii) the OpenFlow Protocol that offers a standard for communication between an outer controller and a switch. OpenRoads or OpenFlow Wireless[62, 63] are recent wireless extensions of OpenFlow.

RcUBe is similar in its objectives to each of the aforementioned approaches. However, it departs from existing work in several ways. First, RcUBe enables real-time reconfigurability at the PHY, MAC, and network layers of the protocol stack. It is based on the software-defined philosophy at the PHY. The current version is based on the GNU Radio framework and USRP front-end to reach a broader extent of PHY layer designs compared to approaches limited to the capabilities of the selected commodity WLAN card [36]. RcUBe offers a strict separation between decision, control and data logic to obtain more flexibility compared to [50]. As mentioned before, it uses a commercial and low-cost RF front-end, and also offers easier programmability compared to [51]. Moreover, it offers real-time self-optimization through its strictly separated design.

E. RcUBe Architecture

RcUBe is structured into four planes, namely *decision*, *control*, *data*, and *register* plane, as illustrated in Fig. 5. The decision plane consists of user-defined decision algorithms that provide the control logic for real-time protocol optimization, adaptation, and reconfiguration. In the control plane, RcUBe stores the required logic for routing and MAC protocol execution as well as data plane management. Data processing takes place in the data plane, while the regis-

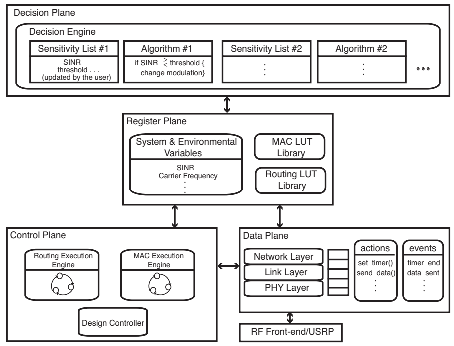


Fig. 5: RcUBe Architecture.

ter plane stores and manages access to system parameters and environmental information. In this section, each plane and their interactions are discussed in detail.

F. Decision Plane

The *decision plane* is the component of RcUBe that provides decisions, at real-time if needed, for the other planes based on user-defined decision algorithms. This plane is designed to be optional upon design, meaning that it can be enabled/disabled for different implementations. The decision plane is interfaced with the control and data planes through the register plane, which acts as a carrier for decisions. This process is described in detail in Section-I..

Decision Engine. The main component of the decision plane is the so-called *decision engine*, which is responsible for the implementation of user-defined decision algorithms. The decision engine contains a set of user-defined algorithms, each associated with a sensitivity list. Each item in the sensitivity list describes conditions under which the corresponding decision algorithm is invoked. Decision algorithms may vary from simple decision rules based on checking the value of a variable, to more complex algorithms requiring adaptive resource allocation as in cognitive radio, or even complex processing based on convex optimization algorithms.

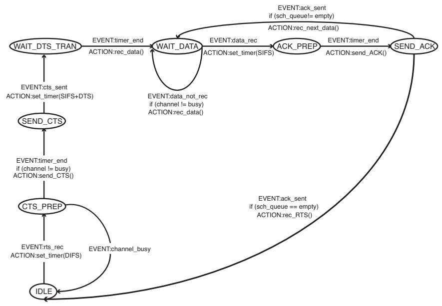
The decision engine is capable of executing multiple decision algorithms in parallel. Algorithms can be executed in a synchronous, or asynchronous fashion, as described in detail in Section-J.. Decision algorithms can (i) modify a parameter in a protocol, (e.g., value of the minimum contention window (CW) in a CSMA protocol), (ii) switch among different modes within a protocol, (e.g., change the modulation scheme), (iii) switch among different protocols altogether. Decisions are taken in the decision plane by the decision engine, but they are not applied to the system by the decision engine itself. After decision making, the decision engine writes the decision result in the register plane. Therefore, decisions can be accessed by the data and control planes to be adopted in their execution logics. This will be further explained in the following sections.

The execution logic of the decision engine can be illustrated with the following toy examples. First, consider a decision algorithm that is defined to decide the modulation scheme of the PHY layer according to the current SINR value. In this case, SINR values, obtained by the data plane, and user-defined threshold value(s), which are stored in the register plane, are included in the sensitivity list and monitored by the decision plane. When the measured SINR value decreases below the level of the user-defined threshold value, the decision engine triggers the execution of the decision algorithm to select the appropriate modulation scheme. The selected modulation scheme parameter is afterwards written into the register plane and thus becomes accessible by the data plane. Subsequently, the modulation scheme parameter will be accessed by the data plane to adapt execution at the physical layer accordingly. Second, consider a network with centralized control. The central node (access point or base station) runs a decision algorithm that adaptively selects a specific MAC protocol and imposes it to the client nodes through control packets, based on the offered load to the network. For example, when the offered load is above a certain value, Time-Division Multiple Access (TDMA) based protocols like TRAMA [64] are selected, since they give better results in energy efficiency and channel utilization when compared to Carrier Sense Multiple Access (CSMA) based algorithms [64]. The process is executed as in the previous example, and at the last step, the decision is written into a parameter that indicates MAC protocol change. The decision controller of the control plane, which will be explained in the following section, detects the parameter change and loads the newly selected MAC protocol design from the MAC look-up-table (LUT) library of the register plane to the MAC execution engine of the control plane.

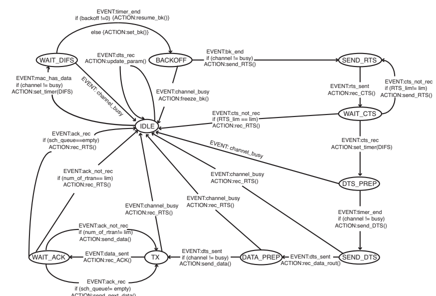
G. Control Plane

The *control plane* is designed to handle the control logic of data processing, which takes place in the data plane. The control plane decides the sequence of data processing functionalities that will be executed in the data plane. It does this mainly by incorporating two execution engines. The two execution engines contain and execute a FSM implementing the logic of either the MAC or the network-layer protocol execution, respectively. The control plane further contains a design controller to manage the logic executed in the execution engines, as shown in Fig.5.

Execution Engines. Execution engines are designed to execute FSMs that define MAC and routing protocols. The FSMs are designed as Extended Finite State Machines. Each FSM contains symbolic-states and an extended state transition definition. Each state transition is designed as a triplet that contains *events*, *conditions*, and *actions*. Each state transition occurs if *events* take place while *conditions* are fulfilled. As a result of that, the corresponding *actions* are invoked. *Actions* are mainly designed to invoke a data processing functionality in the data plane, while *events* may occur as a result of data processing actions.



(a) Receiver path of the FSM.



(b) Transmitter path of the FSM.

Fig. 6: Sample FSM architecture taking place in the MAC execution engine at the control plane. State transitions are based on the triplet (events, conditions, actions).

Conditions are controlled via the register plane, where system parameters and environmental variables are stored. To simplify the definition of new protocols, the state transitions of FSMs are defined by LUTs, which are stored in the register plane to create a library of different protocols, and loaded on the execution engines to be transformed into FSMs for execution. A sample list of *events, conditions, and actions* are shown in Table-1. Each component and its properties are explained in detail in the following sections. The core design idea of using FSM structures is adopted from [36]. Here, it is reformulated at a different level of abstraction to fit our design and to include both MAC and routing protocols. Figure 6 depicts a sample FSM of a MAC layer design described for the CR scenario at [43].

Design Controller. This component is responsible for control and management of the process of loading routing and MAC layer protocols, which are defined by the user as LUTs in the register plane, into the execution engines. It also controls switching between different protocols (pre-defined in the libraries at the register plane) on-the-fly based on the decisions taken by the decision plane. The design controller monitors parameters in the register plane that identify which protocols need to be executed by the execution engines. These parameters can either be updated in real time by decisions taken in decision plane, or offline by the system designer.

Table 1: List of *Events, Conditions, and Actions*.

Event	Condition	Action
mac_has_data	channel!=busy	set_timer()
channel_busy	backoff!=0	resume_bk()
timer_end	backoff==0	set_bk()
bk_end	RTS_lim!=lim	freeze_bk()
bk_end	RTS_lim==lim	send_RTS()
rts_sent	num_of_rtran!=lim	rec_RTS()
dts_sent	num_of_rtran==lim	send_CTS()
cts_sent	sch_queue!=empty	rec_CTS()
ack_sent	sch_queue==empty	send_DTS()
channel_idle	queues!= empty	set_optimization_enable()
data_sent		rec_DTS()
cts_rec		send_data()
cts_not_rec		rec_data()
ack_rec		send_ACK()
ack_not_rec		rec_ACK()
cts_not_rec		send_next_data()
ack_not_rec		send_data()
ack_not_rec		rec_RTS()
data_not_rec		send_next_data()
mac_data_req		send_data_mac()
optimization_end		set_rout_has_data()
mac_has_data		rec_data_mac()
mac_data_rec		identify_data()

H. Data Plane

The *data plane* is where data processing and data/control queues are handled. It also contains two libraries of pre-defined primitive building blocks, *actions* and *events*, and interacts with the RF front-end/USRP. Briefly, data coming from upper layers are received at the network layer, and after header formation at the network and MAC layer, are transformed into digital waveforms at the PHY layer, which is implemented in GNU Radio. Subsequently, digital waveforms are converted into analog waveforms by the digital-to-analog converter (DAC) in the RF front-end.

The data plane has three main design features that separate it from traditional systems. First, *it decomposes protocols into primitive building blocks*. In this way, the data plane has a modular and flexible structure and can be reconfigured by changing the sequence of execution of these primitive building blocks. Primitive building blocks include *actions* and *events*. *Actions* are functional blocks that execute data processing, whereas *events* are indicator blocks for acquiring results of data processing. Second, as briefly discussed in Section-G., is the *separation between data processing and its control logic*. This means that the control plane only decides the sequence of data processing functionalities with its executing engines, while data processing itself takes place in the data plane. Thanks to this separation, RcUBe is able to define different protocols just by changing the sequence of data processing actions. For instance, suppose the control plane decides to send an RTS packet to a neighbor node by invoking the *action* named *"send_RTS()"*. After this, the data plane acquires the next hop information from the routing table stored in the register plane and accordingly forms the RTS packet structure at the MAC layer and transmits it at PHY layer by using the corresponding GNU Radio flowgraph. Note that all the parameters of protocols at different layers are stored as variables in the register plane. Therefore, any parameter of any protocol at any layer can be reconfigured on-the-fly through the register plane. Protocol variables can be accessed "intrinsically" by the decision plane as a result of a decision algorithm, or by the data plane according to specific data acquisitions, or "extrinsically" by the protocol designer.

Actions. *Actions* are primitive building blocks formed as a decomposition of MAC and network layer protocols. Actions are mainly designed to handle data plane processes. Nevertheless, they are also used to update parameters stored in the register plane and to provide synchronization between MAC and routing FSMs, particularly to exchange data between each other. They can be classified based on their functionalities as (i) actions to transmit/receive data and control packets (e.g., ACK, CTS, RTS, among others), including processing of packets with headers and selecting corresponding PHY flowgraph as subsystems, (ii) actions to reconfigure parameters of protocols at different layer, (iii) timing functionalities, (iv) routing functionalities (e.g., forwarding, exchanging route discovery messages), (v) data transfer between different layers.

Events. *Events* are primitive building blocks that are responsible for indicating the results of data processing functions, so they maintain state transitions and protocol executions, as discussed in Section-G..

To handle MAC and routing layer processing, we have built our primitive building blocks as *actions*. Moreover, for routing protocols, we have defined a routing table in the register plane. For PHY layer processing, since our design was developed on USRP, we have taken advantage of the open source GNU Radio framework. GNU Radio provides an extensive library of signal processing blocks. Although many blocks already exist, new blocks can be added if needed. Thus, one can define a PHY protocol simply by connecting these blocks so as to build a *flowgraph*. Thanks to this granularity, our framework can enable reconfigurability in three main ways. First, according to the PHY-layer preferences of the MAC protocol, the control plane can switch between pre-defined flowgraphs. Second, the parameters of blocks that form flowgraphs can be updated through the register plane based on the decision engine. Finally, users can simply connect different blocks to implement new PHY protocols.

I. Register Plane

The *register plane* is responsible for saving both environmental variables/readings (e.g., SINR) and state variables at the different layers (e.g., modulation, transmit power, size of minimum CW, among others). It acts as a reference plane for the other planes (i.e., decision, data, control), and it can be updated by both the decision plane (based on results from executed algorithms) and by the data plane according to environmental sensed data or incoming information from neighboring nodes. It can be accessed by all planes. The register plane also hosts two libraries of LUTs that encapsulate MAC and routing protocols. These libraries are designed to store custom MAC and routing protocols. Therefore, users can easily create their own cross-layer protocol libraries, thus achieving fast prototyping. Moreover, the control plane may use these libraries to switch protocol on-the-fly based on decisions taken at the decision plane. The register plane also hosts neighbor and routing tables, which may be accessed and updated by the decision and data planes depending on specific protocol designs.

J. Architectural Considerations

Delay. The current version of the RcUBe framework is fully implemented in software and it mainly runs on the host PC. This design choice was made to fully leverage the flexibility of the GNU Radio framework for PHY layer processing. As discussed and observed before, pure software implementations of the protocol stack offer great flexibility at the expense of higher processing latency and coarser control on the timing of operations [35, 65]. In Section L., we also observed delays that prevent our framework from meeting time-critical deadlines of the implemented MAC protocols. Thus, as a temporary fix, we defined and adopted specialized guard times to maintain the timing and execution of the MAC protocols. The software-hardware implementation boundaries of our framework are flexible, and we are currently working to shift portions of the execution engines to a soft-core processor implemented on FPGA, thus combining the flexibility offered by high-level programming languages with the processing efficiency of hardware implementations.

Control and Data Packet Handlers. RcUBe offers data and control packet handlers through actions defined in the control plane. The functionalities of these actions are two-fold. Firstly, they provide switching between different physical layer receiver flowgraphs depending on the packet type (i.e., control or data packet). Secondly, they are responsible for the implementation of MAC layer functionalities. For example, consider the action named *"rec_RTS()"* . This action is defined for handling both in physical and MAC layers, control packets named Request-to-Send (RTS).

Asynchronous or Synchronous Decisions. RcUBe supports both asynchronous and synchronous decisions. A decision algorithm can be either executed synchronously, i.e., triggered by another plane, or asynchronously. In particular, for synchronous execution, a flag variable should be defined into the sensitivity list of the corresponding algorithm in the decision plane. Depending on which protocol is adopted in the register plane the activation time of this flag by the control plane is affected. After execution of the corresponding decision algorithm, another flag variable should be enabled to inform the decision plane about completion of execution. However, sometimes the synchronous approach may be harmful for the protocol stack performance because of the delays associated with execution of decision algorithm. In these cases, decision algorithms can be executed asynchronously without expecting an acknowledgement from the decision plane. In this way, the rest of the system does not need to wait for the optimized parameters obtained by the decision algorithm to execute the protocol stack. Therefore, one can either choose to use asynchronous decisions to meet hard deadlines in protocol stack execution and operate with non-optimized parameters, or wait for optimized parameters at the price of extra delay.

Centralized and Distributed Control. RcUBe can be used in networks with both centralized and distributed control. In Section-K., we will discuss the implementation of a complex protocol in a network with distributed control. For networks with centralized control, the following toy example can be explanatory. Consider a cluster-based network with TDMA scheduling. Here, all decisions, e.g., number of time slots to be allocated to each user, are taken by the cluster head and forwarded to cluster members through control packets. At cluster members, RcUBe can be configured with a disabled decision plane. In this way, children nodes can only decode received control packets and write enclosed information (number of slots assigned to them for next session) to their register plane. At the cluster head, an algorithm running in the decision plane assigns time slots to different nodes.

K. Example Design Implementation

Our primary objective is to demonstrate the flexibility, real-time reconfigurability, and self-optimization capabilities offered by RcUBe. To this end, we implemented within the RcUBe framework a joint ROuting and Spectrum Allocation algorithm (ROSA) algorithm [43]. The ROSA algorithm was chosen because of its ability to demonstrate and take advantage of all the proposed features of the RcUBe framework. ROSA is based on a cross-layer architecture (PHY, MAC, and Routing) with complex interactions and employs a decentralized optimization algorithm. Moreover, it continuously relies on real-time decisions and reconfigurations as it adapts its PHY, MAC, and routing behavior according to time-varying traffic demands, network topology, and interference profile. Additionally, on the control channel ROSA employs an IEEE 802.11-like MAC protocol, thus giving us an opportunity to prove that RcUBe can support both conventional, current state-of-the-art protocol designs as well as newly proposed, novel cross-layer protocols.

Definition of the Algorithm. ROSA [43] is a CR algorithm that tries to maximize the network throughput through cross-layer jointly optimal routing, dynamic spectrum allocation, scheduling, transmit power control and interference avoiding waveform selection.

The algorithm was developed specifically for secondary users of a cognitive

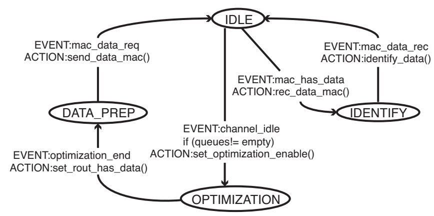
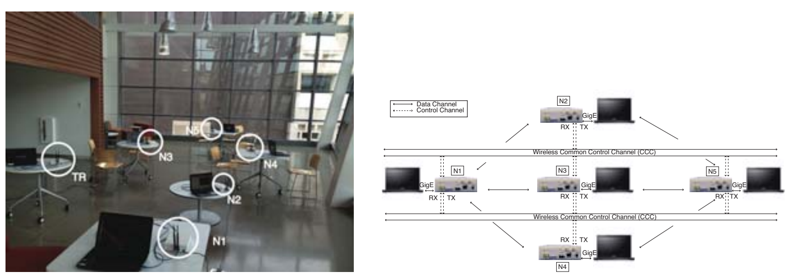


Fig. 7: Routing execution engine for ROSA.

ad hoc networks, which send their data opportunistically without deteriorating on-going (primary) communications. ROSA is based on two different channels. Secondary users contend for spectrum access and exchange local control information on the *common control channel* (CCC), while the *data channel* (DC) consists of equally-sized discrete frequency bands to be allocated by users. ROSA employs an IEEE 802.11-like protocol with three-way handshaking and spectrum sensing for enabling Collaborative Virtual Sensing (CVS) [43]. Each node can acquire local information through explicit exchange of control messages, by overhearing control packets, or through spectrum sensing. Therefore, each backlogged node is able to acquire spectrum and queuing information from its neighbors and can accordingly select the next hop and spectrum band to be used for transmission. An additional control packet, named Data Transmission reServation (DTS), was defined, in addition to Request-to-Send (RTS) and Clear-to-Send (CTS), to be responsible for informing the neighbor nodes about spectrum reservation and transmit power.



(a) Testbed deployment in EE Dept.

(b) Schematic overview of the 5-node testbed setup.

Fig. 8: A view from our testbed implementation.

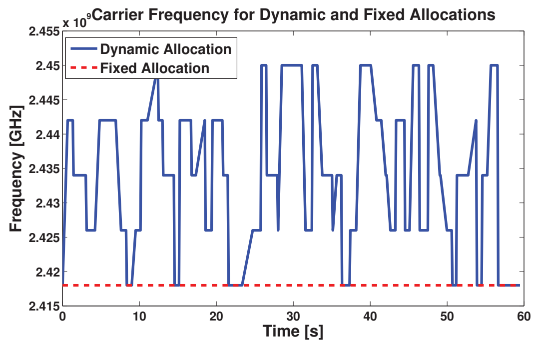
ROSA is based on a backpressure principle and performs a joint routing, dynamic spectrum allocation, and waveform selection algorithm that is triggered whenever a backlogged node senses the CCC to be idle. Each node, source or intermediate node, maintains a separate queue for each session, defined by a fixed node-destination pair. The algorithm identifies a set of feasible next hops for each session and for each of them solves a link capacity maximization problem to determine optimal spectrum and power allocations. Subsequently, it calculates a so-called *spectrum utility function*, defined as the product of maximum link capacity and maximum differential backlog on that link. It then selects session and link (next hop) with maximum spectrum utility function. According to the link chosen, it decides on a modulation scheme (i.e., BPSK, QPSK) based on the current estimated SINR on the chosen channel at the transmitter node. Finally, the node uses the relative spectrum utility function to select the size of the CW, which determines the priority of being scheduled for transmission as compared to neighboring competing nodes.

Interpretation of the Algorithm. The ROSA algorithm and its functionalities are implemented in the RcUBe architecture as follows. The control plane contains the execution engines for routing and MAC protocols that are illustrated in Fig. 7 and Fig. 6 respectively. The routing protocol FSM is designed in such a way as to trigger the decision algorithm that takes place in the decision plane. Moreover, it also invokes *actions* to form and handle network layer headers, in coordination with the MAC-layer FSM, through the routing information contained in the routing table in the register plane. Therefore, the routing FSM prepares packets to be sent to the MAC layer or identifies received packets from the MAC layer. The FSM design of the MAC execution engine is shown in Fig. 6 with two subfigures depicting receiver and transmitter path that are connected to each other through the shared state *IDLE*. This FSM is formed based on common primitive building blocks that can be used for any 802.11-like MAC protocol.

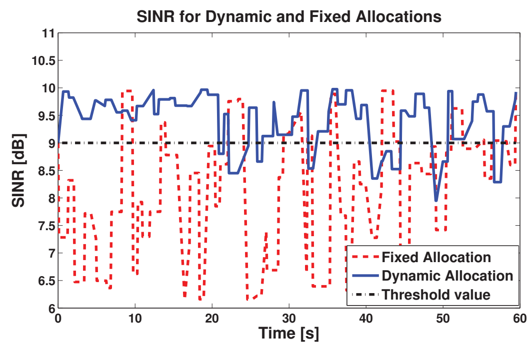
The decision plane hosts and executes the decision algorithm of ROSA that decides next hop, spectrum portion to be occupied, waveform and size of contention window. In order to trigger the execution of this algorithm, the decision engine monitors an enable parameter (*"optimization_enable"*) that is defined in its sensitivity list. This flag is raised as a consequence of an action invoked by the control plane (routing execution engine) and stored in the register plane. After execution of the decision algorithm, the updated, optimized parameters are written in the register plane. Moreover, an event is raised so as to indicate the end of the decision algorithm execution. This event is used as a state-transition trigger in the routing execution engine. This process is a good example of synchronous decisions, as discussed in Section J.. The data plane contains the data processing of PHY, routing layers and queues (sessions) for data packets. The register plane is the storage space for all the system and environmental parameters that can be accessed by other planes. Apart from that, it also hosts the neighbor table that is pre-defined for the ROSA algorithm, routing table, and LUT libraries for other MAC and routing protocols.

L. Experimental Evaluation

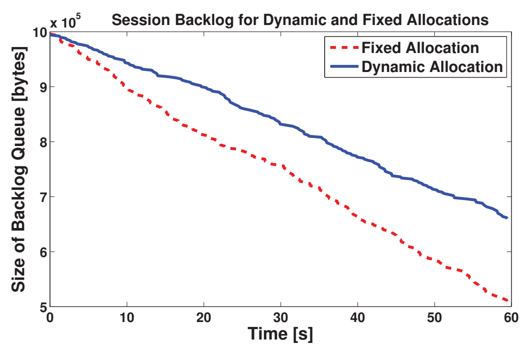
In this section, we present experimental results obtained by implementing the ROSA protocol suite in RcUBe in our testbed depicted in Fig. 8-a. Each wireless



(a) Carrier Frequency for Dynamic and Fixed Allocations.



(b) SINR for Dynamic and Fixed Allocations.



(c) Session Backlog for Dynamic and Fixed Allocations.

Fig. 9: Testbed Results for 2-Node Setup.

software-defined-radio (SDR) node consists of a Linux-PC, hosting the RcUBe framework, connected via Gigabit Ethernet (GigE) with a USRPN − 210 SDR device, developed by Ettus [52]. Each PC is equipped with an Intel i7 − 3610QM 2.30 GHz processor, 8 GByte RAM. The operating system is Ubuntu Precise Pangolin 12.04. We used up to 5 nodes named N1, N2, N3, N4, N5, while an extra 6th node (named TR) was used for capturing trace data for offline processing.

We used an orthogonal frequency-division multiplexing (OFDM) scheme with 200 subcarriers at the PHY layer. Each OFDM subcarrier is modulated either with Binary-Phase-Shift-Keying (BPSK) or Quadrature-Phase-Shift-Keying (QPSK). In each communication link, the modulation scheme is picked according to a decision made by the algorithm based on the SINR estimation of this link. Specifically, we defined a SINR threshold at 9 dB. Therefore, if the SINR estimate of the chosen link is lower than this threshold, the algorithm selects BPSK, otherwise it selects QPSK. We considered a CCC, which has a center frequency at 2.41 GHz with a bandwidth of 1 MHz. We considered a DC with five channels with center frequencies at 2.418, 2.426, 2.434, 2.442, and 2.450 GHz respectively, and 1 MHz bandwidth.

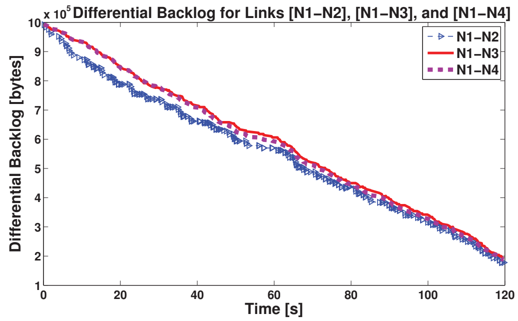
We considered two networks of different size, i.e., 2-Node and 5-Node setups, to focus on different features and capabilities of the framework.

2-Node Setup. The first set of experiments were implemented on a 2-Node setup, where there was 1-session defined from N1 to N2. In these experiments, our intention was to "degrade" the ROSA decision algorithm to a dynamic spectrum allocation and optimal waveform selection algorithm only, to concentrate on PHY reconfigurations. Since we had only one hop, the differential backlog factor was not considered and the ROSA algorithm simply determined the best spectrum allocation and waveform selection to maximize the channel capacity. In this set of experiments, we considered two different scenarios. In the first one we used dynamic spectrum allocation as in ROSA, while in the second we used fixed spectrum allocation. The algorithm just decides the modulation scheme to be picked based on the SINR estimate. Each experiment lasted for 60 s.

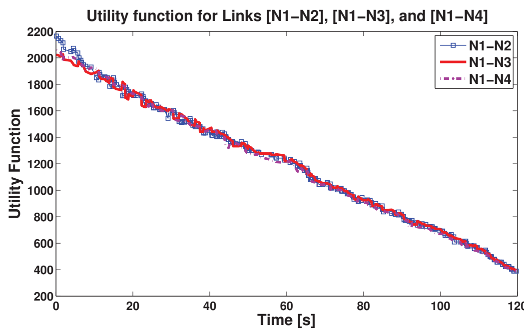
Figure 9 illustrates the experimental results acquired by the setup described above. Figure 9-(a) depicts the frequency allocation results for both cases (dynamic and fixed allocation), while Fig. 9-(b) illustrates the corresponding SINR estimates. It is clear that dynamic spectrum allocation obtains a higher SINR profile when compared to fixed spectrum allocation. As a result, dynamic spectrum allocation can obtain higher transmission rates as it can select higher-order modulation schemes (i.e., QPSK instead of BPSK). This effect can be observed in Fig 9-(c), which shows the size of session backlog for both cases. It can be concluded that when spectrum is allocated dynamically, session backlog gets cleared faster than with fixed spectrum allocation. Overall, real-time decisions about the PHY layer parameters (i.e., modulation and carrier frequency selection) can be applied in real time.

5-Node Setup The second set of experiments was implemented on a 5-Node setup, where there was 1-session defined from N1-N5, active for 120 s. Based on ROSA, each node has a pre-defined neighborhood table for routing purposes. We defined the neighborhood in such a way that N2, N3 and N4 were intermediate nodes, only one hop away from destination node N5. Each node can overhear control packets from neighbors through the CCC. The testbed schematic in Fig. 8-(b) shows the neighborhood connections between the nodes.

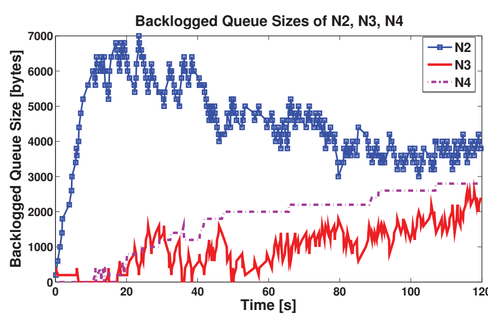
Figures 10 and 11 depict the results acquired from the experiments conducted



(a) Differential Backlog for Links [N1-N2], [N1-N3], and [N1-N4].



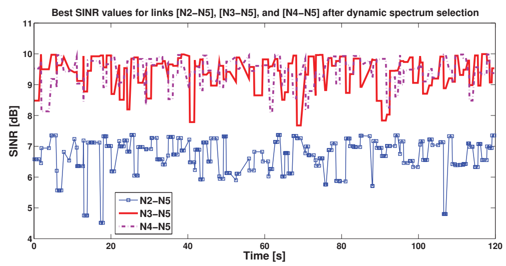
(b) Utility Function for Links [N1-N2], [N1-N3], and [N1-N4].



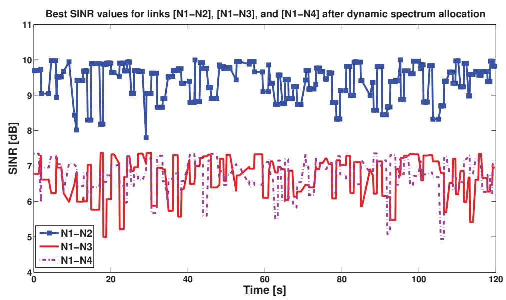
(c) Backlogged Queue Sizes of N2, N3, and N4.

Fig. 10: Testbed Results for 5-Node Setup with 1-Session at [N1-N5].

in the 5-Node setup. Figure 10-(a) illustrates the differential backlog for the links between the source and three intermediate nodes. During the first 10 seconds, N2 gets most of the traffic load from N1 as its link has higher SINR compared to N3 and N4. This can also be observed in Fig. 11. Accordingly, higher spectrum utility is achieved because of the higher channel capacity. Moreover, since link N1-N2 has high SINR values, it uses QPSK modulation, which results in higher



(a) Best SINR values for links [N2-N5], [N3-N5], and [N4-N5] after dynamic spectrum selection.



(b) Best SINR values for links [N1-N2], [N1-N3], and [N1-N4] after dynamic spectrum selection.

Fig. 11: Best SINR values for links in the 5-Node Setup with 1-Session at [N1-N5].

transmission rates. Between the 10th and 20th second, consequent to its fast start, N1-N2 reaches a lower differential backlog value compared to the other links, because of backlogged packets at N2. Therefore, the utility functions of the two links are balanced, as can be observed from Fig. 10-(b), and eventually the traffic is shared equally between the intermediate nodes. After the 20th second, it can be observed that each one hop link has similar utility function, which makes them receive incoming traffic from the source equally. However, it can be observed from Fig. 10-(a), that the differential backlog value of the N1-N2 link reaches the level of others around the 120th second. The reason behind this can be understood by observing Fig. 10-(c), where the backlog queue sizes of intermediate nodes are shown. The high traffic offered from the N1-N2 link and the low SINR values at the N2-N5 link cause N2 to become highly backlogged. However, after the 20th second, as N2 is more likely to be scheduled for transmission due to the narrow contention window selection (which is directly related to the utility function of the link) the number of backlogged packets decreases. Therefore, the N1-N2 link has a higher differential backlog than before, even if it did not get more traffic.

It can be concluded that each node was capable of taking decisions and reconfigure in real-time PHY (i.e., waveform selection, spectrum allocation), MAC (i.e., size of CW) and network layer parameters (i.e., next hop decisions) through a decision algorithm.

M. Software Defined Radio All-spectrum Cognitive Channelization

Cognitive Radio (CR) has emerged as a promising technology for improving spectrum utilization efficiency [66, 67, 68, 69]. CR systems have been envisioned as intelligent, flexible, radio systems capable of reconfiguring their own parameters (i.e., modulation, waveform, carrier frequency etc.) in an autonomous way [70], and may compare favorably to traditional communication systems which are missing such capabilities and often criticized for wasting resources. The evolution of software defined radio (SDR) technology enables implementation of such "smart" radios, capable of reconfiguring crucial communication parameters in software, offering that way extreme agility over hardware radios with fixed characteristics. CR users, also called secondary users, may coexist with primary users (spectrum licensees) and be allowed to access licensed spectrum provided that their interference to primary users is maintained below a prespecified threshold (transmission in grey spaces [72, 71, 73, 75, 74]). Alternatively, CR users may transmit in holes (white spaces) of unutilized spectrum which may not be the preferred accessing method in highly occupied spectrum bands [76]. SDR implementation of cognitive radios has been considered in [78, 79], where secondary users sense the spectrum and transmit in spectrum holes that may or may not be contiguous.

Over the past several years we have been working on and contributing to cognitive radio technologies, where primary and secondary users coexist and utilize the same spectrum concurrently. In this project, we consider secondary links coexisting with unknown primary users and we experimentally evaluate the CR spectrum access methodology proposed in [72, 71, 73, 75, 74] on a low-cost, indoors deployed SDR testbed. Spectrum access is achieved through adaptive channelization where adaptively optimized waveforms are assigned to secondary users. Waveforms are designed to maximize the output signal-to-interference plus noise ratio (SINR) at the output of the secondary receivers, and at the same time, maintain the interference to the primary users below a prespecified threshold [72, 71, 73, 75, 74, 77].

Our testbed is deployed in an RF-cluttering lab environment, assumes unknown primary users and utilizes low-cost, commodity transceivers whose internal oscillators are quite unstable, and thus introduce carrier frequency offsets (CFOs) between any transmitter-receiver pair. Thus, the waveform design problem presented in [73], has to be coupled with estimation of the unknown multipath channel coefficients and carrier frequency offsets. In other words, while our main objective in this project is the experimental evaluation of the CR access methodology in [72, 71, 73, 75, 74], as a side theoretical result we develop a technique for blind estimation of multipath channel coefficients and CFOs. More specifically, we propose a generalized version of the subspace-based estimation procedure presented in [80]. Then, using three commercial SDR transceivers, we experimentally validate the concept of cognitive channelization and spectrum sharing between primary and secondary users, in terms of instantaneous SINR and bit-error-rate (BER).

N. System Model

In this work, we consider a low-cost secondary link (secondary transmitter-receiver pair) operating in the same frequency as a primary user. Both the secondary and primary users are assumed to transmit binary antipodal information symbols $b_k(i) \in \{-1, 1\}$, $i = 0, \ldots, J-1$ and $k = 1, 2$ at a rate $1/T$, modulated by a signal waveform $d_k(t)$ of duration T. If f_c denotes the common carrier frequency, then the transmitted signal of the secondary user can be expressed as

$$x_s(t) = \sum_{i=0}^{J-1} b_1(i)\sqrt{E_1} d_1(t - iT) e^{j(2\pi f_c t + \phi_1)} \tag{39}$$

while the transmitted signal of the primary user is given by

$$x_p(t) = \sum_{i=0}^{J-1} b_2(i)\sqrt{E_2} d_2(t - iT) e^{j(2\pi f_c t + \phi_2)} \tag{40}$$

where E_k, $k = 1, 2$ denotes transmitted energy per bit and ϕ_k, $k = 1, 2$ is the carrier phase relative to the transmitter's local oscillator. Information bits $b_k(i)$, $k = 1, 2$ are modulated by unique digital waveforms $d_k(t)$, $k = 1, 2$, respectively given by

$$d_k(t) = \sum_{l=0}^{L-1} s_k(l) g_T(t - lT_d), \quad k = 1, 2 \tag{41}$$

where $s_k(l) \in \{-1, 1\}$ is the l-th waveform-bit of the waveform assigned to the k-th transmitter, $k = 1, 2$, $g_T(\cdot)$ is a pulse shaping square-root raised cosine (SRRC) filter of duration T_d, and $T = LT_d$ is the waveform duration (period).

All signals are considered to propagate over Rayleigh multipath fading channels and experience complex additive white Gaussian noise (AWGN) at the receiver. Multipath fading is modeled by a linear tapped-delay line with taps that are spaced at T_d intervals and are weighted by independent fading coefficients. The received signal in the presence of multipath fading is bandlimited to $1/T_d$ and can be expressed by

$$r_c(t) = \sum_{n=0}^{N-1} h_{1,n} x_s(t - nT_d) + \sum_{n=0}^{N-1} h_{2,n} x_p(t - nT_d) + n(t) \tag{42}$$

where N denotes the total number of resolvable paths (assumed to be the same for both the primary and secondary transmitter), $h_{k,n}$, $k = 1, 2$, $n = 0, \ldots, N-1$, are independent zero-mean complex Gaussian random variables that model the fading phenomena and remain constant over T_c. $\mathbf{n}(t)$ denotes a complex white Gaussian noise process.

We note that low-cost commodity transceivers introduce carrier frequency offsets $\{\Delta f_k\}$, $k = 1, 2$, between any transmitter-receiver pair. Therefore, the received baseband signal after carrier demodulation is given by

$$r(t) = \sum_{i=0}^{J-1} \sum_{k=1}^{2} b_k(i) \times \sum_{n=0}^{N-1} \tilde{h}_{k,n} d_k(t - iT - nT_d) e^{-j2\pi\Delta f_k t} + n(t) \tag{43}$$

where $\tilde{h}_{k,n} = \sqrt{E_k} h_{k,n} e^{-j(2\pi f_c n T_d - \phi_k)}$ (the resulting carrier phase is absorbed in the channel coefficient). After pulse matched filtering and sampling of the received signal $r(t)$ at rate $1/T_d$ over the multipath extended symbol period of $L + N - 1$ waveform-bits, the received data

$$\mathbf{r} = \begin{bmatrix} r(0) & r(T_d) & \dots & r((JL + N - 2)T_d) \end{bmatrix}^T \in \mathbb{C}^{(JL+N-1)\times 1} \tag{44}$$

become ready for filtering and detection of the information bits of the secondary user $b_1(i)$, $i = 0, \dots, J - 1$. With respect to the i-th bit, we can re-write the received data vector in the following form

$$\mathbf{y}_i = [\mathbf{r}]_{(0+iL)T_d:T_d:((L+N-2+iL)T_d)} \tag{45}$$

which is equivalent to

$$\mathbf{y}_i = b_1(i)\mathbf{\Gamma}_1(i, \Delta f_1)\mathbf{H}_1\mathbf{s}_1 + \mathbf{p}_i + \mathbf{n}_i, \qquad i = 0, \dots, J - 1 \tag{46}$$

where $\mathbf{\Gamma}_1(0, \Delta f_1) \stackrel{\triangle}{=} \mathrm{diag}\big(1, \mathrm{e}^{-\mathrm{j}2\pi\Delta \mathrm{f}_1 \mathrm{T_d}}, \dots, \mathrm{e}^{-\mathrm{j}(2\pi\Delta \mathrm{f}_1(\mathrm{N+L-2})\mathrm{T_d})}\big)$ is a $(N + L - 1) \times (N + L - 1)$ diagonal matrix consisting of the frequency offset values between the secondary transmitter and receiver, and $\mathbf{H}_1 \in \mathbb{C}^{(L+N-1)\times L}$ is the multipath fading channel matrix given as follows

$$\mathbf{H}_1 = \sum_{n=0}^{N-1} \tilde{h}_{1,n} \begin{bmatrix} \mathbf{0}_{n\times L} \\ \mathbf{I}_{L\times L} \\ \mathbf{0}_{(N-n-1)\times L} \end{bmatrix}. \tag{47}$$

We assume that $\mathbf{n}_i \sim \mathcal{CN}(\mathbf{0}, \sigma^{\mathbf{2}}_{\mathbf{n}}\mathbf{I}_{\mathbf{N+L-1}})$ represents complex zero-mean white Gaussian noise, and $\mathbf{p}_i$ accounts for the unknown primary user signal. The frequency offset values in $\mathbf{\Gamma}_1(i, \Delta f_1)$ remain constant over T_c symbols.

O. Optimal Waveform Design

The cumulative interference for the secondary user signal of interest in (46) is $\mathbf{p}_i + \mathbf{n}_i$. If Δf_1 and the channel coefficients $\tilde{h}_{1,0}, \tilde{h}_{1,1}, \dots, \tilde{h}_{1,N-1}$ are known then the linear filter that operates on $\mathbf{y}_i$ and maximizes the SINR at its output is given by

$$\mathbf{w}_{\text{maxSINR}} = \arg\max_{\mathbf{w}} \frac{\mathrm{E}\{|\mathbf{w}^{\mathrm{H}}(\mathrm{b}_1\mathbf{\Gamma}_1(\mathrm{i},\Delta\mathrm{f}_1)\mathbf{H}_1\mathbf{s}_1)|^2\}}{\mathrm{E}\{|\mathbf{w}^{\mathrm{H}}(\mathbf{p}+\mathbf{n})|^2\}}$$

$$= (\mathbf{R}_p + \sigma_n^2\mathbf{I}_{N+L-1})^{-1}\mathbf{\Gamma}_1(i,\Delta f_1)\mathbf{H}_1\mathbf{s}_1. \tag{48}$$

If we denote the cumulative disturbance autocorrelation matrix as $\mathbf{R}_{I+N} = \mathbf{R}_p + \sigma_n^2\mathbf{I}_{N+L-1}$, then the maximum output SINR value attained at the secondary receiver is given by

$$\mathrm{SINR}_{\max} = \frac{\mathrm{E}\left\{|\mathbf{s}_1^{\mathrm{T}}\mathbf{H}_1^{\mathrm{H}}\mathbf{\Gamma}_1^{\mathrm{H}}(\mathrm{i},\Delta\mathrm{f}_1)\mathbf{R}_{\mathrm{I+N}}^{-1}(\mathrm{b}_1\mathbf{\Gamma}_1(\mathrm{i},\Delta\mathrm{f}_1)\mathbf{H}_1\mathbf{s}_1)|^2\right\}}{\mathrm{E}\left\{|\mathbf{s}_1^{\mathrm{T}}\mathbf{H}_1^{\mathrm{H}}\mathbf{\Gamma}_1^{\mathrm{H}}(\mathrm{i},\Delta\mathrm{f}_1)\mathbf{R}_{\mathrm{I+N}}^{-1}(\mathbf{p}+\mathbf{n})|^2\right\}}$$

$$= \mathbf{s_1}^T\mathbf{H}_1^H\mathbf{\Gamma}_1^H(i,\Delta f_1)\mathbf{R}_{I+N}^{-1}\mathbf{\Gamma}_1(i,\Delta f_1)\mathbf{H}_1\mathbf{s}_1. \tag{49}$$

Now, we consider $\mathrm{SINR}_{\max}$ in [76] as a function of the waveform $\mathbf{s}_1$. In this project, we propose to design $\mathbf{s}_1^{\text{opt}}$ such that it maximizes the SINR at the output of the maximum SINR receiver filter. Then, the optimally designed waveform $\mathbf{s}_1^{\text{opt}}$ is fed back by the secondary receiver to the secondary transmitter enabling this way cognitive channelization. Thus, $\mathbf{s}_1^{\text{opt}}$ may be expressed as

$$\mathbf{s}_1^{\text{opt}} = \arg\max_{\mathbf{s}_1}\left\{\mathbf{s_1}^{\mathrm{T}}\mathbf{H}_1^{\mathrm{H}}\mathbf{\Gamma}_1^{\mathrm{H}}(\mathrm{i},\Delta\mathrm{f}_1)\mathbf{R}_{\mathrm{I+N}}^{-1}\mathbf{\Gamma}_1(\mathrm{i},\Delta\mathrm{f}_1)\mathbf{H}_1\mathbf{s}_1\right\}. \tag{50}$$

Let us now define $\tilde{\mathbf{R}}(\Delta f_1) \stackrel{\triangle}{=} \mathbf{H}_1^H\mathbf{\Gamma}_1^H(i,\Delta f_1)\mathbf{R}_{I+N}^{-1}\mathbf{\Gamma}_1(i,\Delta f_1)\mathbf{H}_1$. Let the eigenvectors of the matrix $\tilde{\mathbf{R}}(\Delta f_1)$ are denoted by $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_L$ with corresponding eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_L$, then $\mathbf{s}_1^{\text{opt}}$ in (50) is the eigenvector that corresponds to the maximum eigenvalue λ_1, i.e., $\mathbf{s}_1^{\text{opt}} = \mathbf{q}_1$.

We note that waveform optimization under the constraint of maintaining the performance of the primary user above a presepecified threshold can be carried out in an iterative/coupled fashion where the solution provided by (50) is used to further optimize the power allocated to the secondary user, so that its interference to the primary user is maintained below a prespecified threshold. The coupled optimization problem will be addressed in a continuation of this work, and is not considered here.

P. Joint Subspace-based Channel and CFO Estimation

Expression (50) assumes that $\mathbf{H}_1$, $\mathbf{\Gamma}_1(\cdot,\cdot)$ and $\mathbf{R_{I+N}}$ are known. In practice, we utilize estimates $\hat{\mathbf{H}}_1$, $\hat{\mathbf{\Gamma}}_1(\cdot,\cdot)$ and $\hat{\mathbf{R}}_{I+N}$ based on a finite-size of data/observations. In particular, $\mathbf{R}_{I+N}$ is sample-average estimated [82, 83] over J silent snapshots of the secondary user, i.e.,

$$\hat{\mathbf{R}}_{I+N} = \frac{1}{J}\sum_{i=1}^{J}(\mathbf{p}_i+\mathbf{n}_i)(\mathbf{p}_i+\mathbf{n}_i)^H. \tag{51}$$

The rest of this section deals with the problem of blind estimation of the unknown channel coefficients and frequency offset of the secondary user in the presence of unknown interference and AWG noise. The phase ambiguity induced by blind channel-estimation methods is resolved using a short pilot bit sequence [80].

The proposed method for jointly estimating the unknown channel coefficients $\tilde{h}_{1,n}$ for $n = 0, \ldots, N-1$ and the carrier frequency offset Δf_1, is a modified version of the algorithm presented in [80] where it was shown that the rank of the noise subspace is increased by two-sided truncation of the received signal vector $\mathbf{y}_i$. For our setup, the later implies that trimming the intersymbol interference (ISI) terms for the secondary user from both sides, reduces the signal subspace rank r_s, to $L-N+1 \leq r_s \leq 2K-1$ and offers the maximum possible guaranteed minimum rank of the noise subspace of $(L-N+1)-(2K-1)$.

In this context, let the $(L-N+1)$-length truncated version of the received vector $\mathbf{y}_i$ with respect to the i-th bit be as follows

$$\mathbf{y}_i^{\mathrm{tr}} = \begin{bmatrix} y_i(N-1)T_d & y_i(NT_d) & \ldots & y_i((L-1)T_d) \end{bmatrix}^T \tag{52}$$

which is equivalent to

$$\mathbf{y}_i^{\mathrm{tr}} = \mathbf{b}_1(i)\boldsymbol{\Gamma}_1^{\mathrm{tr}}(i, \Delta f_1)\mathbf{A}_0^s\tilde{\mathbf{h}}_1 + \mathbf{p}_i^{\mathrm{tr}} + \mathbf{n}_i^{\mathrm{tr}} \tag{53}$$

where $\tilde{\mathbf{h}}_1 = \begin{bmatrix} \tilde{h}_{1,0} & \tilde{h}_{1,1} & \ldots & \tilde{h}_{1,N-1} \end{bmatrix}^T$ is the channel coefficients vector of the secondary user, $\boldsymbol{\Gamma}_1^{\mathrm{tr}}(i, \Delta f_1)$ is a $(L-N+1) \times (L-N+1)$ is the truncated version of the diagonal matrix $\boldsymbol{\Gamma}_1(i, \Delta f_1)$ defined in (46), and

$$\mathbf{A}_0^s \triangleq \begin{bmatrix} s_1[N-1] & s_1[N-2] & \ldots & s_1[0] \\ s_1[N] & s_1[N-1] & \ldots & s_1[1] \\ \vdots & \vdots & & \vdots \\ s_1[L-1] & s_1[L-2] & \ldots & s_1[L-N] \end{bmatrix} \tag{54}$$

denotes the $(L-N+1) \times N$ matrix, consisting of shifted versions of the waveform bits of the secondary user, after eliminating $N-1$ elements from both sides of each column.

Let $\mathbf{R}_{\mathrm{tr}} = \mathrm{E}\{\mathbf{y}_{\mathrm{i}}^{\mathrm{tr}}\mathbf{y}_{\mathrm{i}}^{\mathrm{tr}^{\mathrm{H}}}\}$ denote the autocorrelation matrix of $\mathbf{y}_i^{\mathrm{tr}}$. Let $\mathbf{R}_{\mathrm{tr}} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^H$ denote the eigendecomposition of the input autocorrelation matrix, where $\mathbf{Q}$ is a matrix with columns the eigenvectors of $\mathbf{R}_{tr}$ and $\boldsymbol{\Lambda}$ is diagonal matrix with the eigenvalues of $\mathbf{R}_{\mathrm{tr}}$. Using as columns the eigenvectors that correspond to the $(L-N+1)-(2K-1)$ smallest eigenvalues, we form the matrix $\mathbf{U}_n \in \mathbb{C}^{(L-N+1)\times((L-N+1)-(2K-1))}$. These "bottom" eigenvectors of $\mathbf{R}_{\mathrm{tr}}$ belong to the noise subspace, therefore the columns of $\mathbf{U}_n$ correspond to "noise eigenvectors".

We note that when $\mathbf{R}_{\mathrm{tr}}$ is known, the nullspace of $\mathbf{U}_n^H$ coincides with the signal subspace ($r_s = 2K-1$). However, in this work, $\mathbf{R}_{\mathrm{tr}}$ is not known, but rather sample-average estimated by a finite-size record of truncated input data/observations, i.e.,

$$\hat{\mathbf{R}}_{\text{tr}} = \frac{1}{J}\sum_{i=1}^{J}\mathbf{y}_i^{\text{tr}}\mathbf{y}_i^{\text{tr}^{\text{H}}} \tag{55}$$

where J denotes the size of the data record. Thus, $\hat{\mathbf{U}}_n$ is an estimate of the noise subspace, based on the eigendecomposition of the estimated input autocorrelation matrix $\hat{\mathbf{R}}_{\text{tr}}$.

The estimates of $\tilde{\mathbf{h}}_1$ and Δf_1 are found by solving the following constrained optimization problem

$$(\mathbf{\Gamma}_1^{\text{tr}}(i,\Delta f_1)\mathbf{A}_0^s\mathbf{h}_1)^H\hat{\mathbf{U}}_n = \mathbf{0}, \quad \text{subject to} \quad ||\mathbf{h}_1|| = 1. \tag{56}$$

The solution to (56) essentially makes the secondary user signal of interest $\mathbf{\Gamma}_1^{\text{tr}}(i,\Delta f_1)\mathbf{A}_0^s\tilde{\mathbf{h}}_1$, orthogonal to the noise subspace [85]. Propositions 1 and 2 below are modified versions of Propositions 2 and 3 in [80] and are included here for easy reference.

Proposition 1 *Expression* (56) *holds true for values of* Δf_1 *chosen to make the matrix*

$$\mathbf{M}_1(\Delta f_1) \triangleq \mathbf{A}_0^{s^H}\mathbf{\Gamma}_1^{\text{tr}^{\text{H}}}(i,\Delta f_1)\hat{\mathbf{U}}_n\hat{\mathbf{U}}_n^H\mathbf{\Gamma}_1^{\text{tr}}(i,\Delta f_1)\mathbf{A}_0^s \tag{57}$$

singular. Given the choice of Δf_1*, we select* $\tilde{\mathbf{h}}_1$ *as the eigenvector that corresponds to the zero-eigenvalue of* $\mathbf{M}_1(\Delta f_1)$*.*

The following proposition provides a sufficient condition for the uniqueness of the solution provided in Proposition 1.

Proposition 2 *The parameters* Δf_1 *and* $\tilde{\mathbf{h}}_1$ *are uniquely determined if*

(i) $\text{rank}(\mathbf{A}_0^{\text{s}}) = \text{N}$ *and*

(ii) $\text{null}(\hat{\mathbf{U}}_{\text{n}}^{\text{H}}) \cap \text{range}(\mathbf{\Gamma}_1^{\text{tr}}(\text{i},\Delta\text{f})\mathbf{A}_0^{\text{s}})$
$$= \begin{cases} \text{range}(\mathbf{\Gamma}_1^{\text{tr}}(\text{i},\Delta\text{f}_1)\mathbf{A}_0^{\text{s}}\tilde{\mathbf{h}}_1) & \Delta f = \Delta f_1 \\ \{\mathbf{0}\}, & \Delta f \neq \Delta f_1. \end{cases}$$

The proposed estimates $\widehat{\Delta f}_1$ and $\hat{\tilde{\mathbf{h}}}_1$ are then used to form $\hat{\mathbf{w}}_{\text{maxSINR}}$, and $\mathbf{s_1^{\text{opt}}}$ as described in Section O., according to modified versions of (48) and (50).

We note that the blind channel estimate $\hat{\tilde{\mathbf{h}}}_1$ is phase ambiguous, and so is $\hat{\mathbf{w}}_{\text{maxSINR}}$. Phase ambiguity is resolved by applying the mean-square where ϕ is the unknown phase, we apply the mean-square optimum phase recovery method proposed in [80] for correcting the phase of the overall receiver filter $\hat{\mathbf{w}}_{\text{maxSINR}}$. Specifically, the selection criterion of ϕ is based on the minimization of the mean-square error (MSE) between the phase-corrected filter processed data $\hat{\mathbf{w}}_{\text{maxSINR}}^H\mathbf{y}_i$ and the desired information bit $b_1(i)$ for $i = 0, \ldots, J-1$,

$$\hat{\phi} = \arg\min_\phi \text{E}\{|\hat{\mathbf{w}}_{\text{maxSINR}}^{\text{H}}\mathbf{y}_{\text{i}} - \text{b}_1(\text{i})|^2\}, \quad \phi \in [-\pi,\pi). \tag{58}$$

The optimum phase correction according to the optimization criterion (58) is given by (see [80])

$$\hat{\phi} = \text{angle}\left\{\hat{\mathbf{w}}^{\mathrm{H}}_{\mathrm{maxSINR}}\mathrm{E}\{\mathbf{y}\mathrm{b}_1^*\}\right\} \tag{59}$$

where $\mathrm{E}\{\mathbf{y}\mathrm{b}_1^*\}$ can be sample-average estimated by $\frac{1}{P}\sum_{i=1}^{P}\mathbf{y}_i b_1^*(i)$, when a sequence of P pilot information bit symbols is available. Consequently, we recover the phase of the filter estimate $\hat{\mathbf{w}}_{\mathrm{maxSINR}}$, then we correct its phase

$$\hat{\mathbf{w}}_{\mathrm{maxSINR}}e^{j\hat{\phi}}, \quad \hat{\phi} = \text{angle}\left\{\hat{\mathbf{w}}^{\mathrm{H}}_{\mathrm{maxSINR}}\left[\frac{1}{\mathrm{P}}\sum_{\mathrm{i}=1}^{\mathrm{P}}\mathbf{y}_{\mathrm{i}}\mathrm{b}_1^*(\mathrm{i})\right]\right\} \tag{60}$$

and finally, we apply the optimum (in terms of minimizing probability of error) bit detector [86]

$$\hat{b}_1(i) = \mathrm{sgn}\left\{\Re\left\{\hat{\mathbf{w}}^{\mathrm{H}}_{\mathrm{maxSINR}}\mathrm{e}^{-\mathrm{j}\hat{\phi}}\mathbf{y}_{\mathrm{i}}\right\}\right\}, \quad \mathrm{i} = 0, \ldots, \mathrm{J}-1. \tag{61}$$

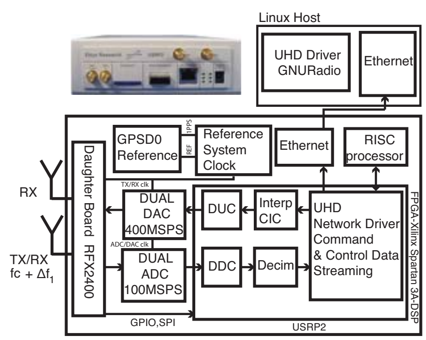


Fig. 12: USRP2 block diagram with attached RFX2400.

Q. Experimental Testbed Setup

In this section, we focus on the software design, implementation in SDR and final deployment of a low-cost indoors testbed for evaluating the concepts of cognitive channelization presented above. For this reason 4 commercial SDR transceivers named Universal Software Radio Peripheral (USRP2) and developed by Ettus [88] are used to provide a setup of both a primary and a secondary link.

R. USRP2 **Receiver: Low-cost, Commercial SDRs**

USRP2 is a low-cost, simple and flexible SDR platform that consists of two 14-bit analog-to-digital converters (ADCs), capable of 100MS/s, two 16-bit digital-to-analog converters (DACs), capable of 400MS/s, a Gigabit Ethernet (GigE) interface and a Xilinx Spartan 3A FPGA for high rate signal processing; all on-motherboard. RF front-end functionality is provided by a wide range of daughterboards which are able to operate from DC to 5.9GHz.

In our experiments, RFX2400 daughterboards were used. These boards offer a homodyne (zero-IF i.e., no intermediate frequency) receiver architecture and operate at $2.25 - 2.9$GHz frequencies. At the receiver path of USRP2, the FPGA attached on the motherboard is responsible for transferring via GigE the incoming from RF front-end and sampled by the ADC, digital baseband data to a host-PC for further signal processing. In the case that USRP2 acts as a transmitter, the on-board FPGA is waiting for PC data which will be sent to the DAC, get converted into analog samples and transmitted via the attached daughterboard. Received baseband signals are sent to the PC in the format of 16-bit in-phase and 16-bit quadrature data, which is equivalent to 4bytes per complex sample. Therefore, the maximum data rate for the baseband IQ data is $\frac{125\text{MB/s}}{4\text{B/Sample}} \simeq 30\text{MS/s}$ (due to overhead 25MS/s). However, the true maximum sampling rate that the PC is capable of handling depends every time on its hardware specifications and data processing capabilities.

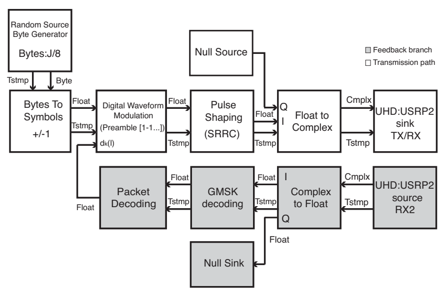


Fig. 13: Secondary transmitter block diagram.

Physical layer processing takes place at the host PC by using GNU Radio [89], a free software development toolkit. GNU Radio provides the signal processing runtime and processing blocks to implement software radios while it controls low-cost external RF hardware such as the USRP2 through the Universal Hardware Driver (UHD). UHD provides a host driver and an application programming interface (API) for USRP2.

GNU Radio offers a plethora of signal processing blocks and libraries offering primitive functionalities i.e., filtering, different kinds of modulation (BPSK, QPSK, GMSK), carrier phase synchronization etc. However, due to the unique and novelty features characterizing the proposed technique of cognitive channelization, we had to build our own custom signal processing blocks and wrap them with the existing interface. Moreover, the timestamp control module offered along with GNU Radio is exploited in our scenario for enabling accurate scheduling between the secondary receiver-transmitter pair. Specifically, input and output streams in both transceivers are tagged with timestamp information, allowing us in that way to control the silent snapshots of the secondary transmitter during which the sample-average estimation of the disturbance autocorrelation matrix $\hat{\mathbf{R}}_{I+N}$ should be done. Extreme attention was also given in the design and implementation of the new signal processing blocks as far as their number of input/output streams and per input data processing is concerned, and how the latter is correlated with the throughput-delay trade-off that each block introduces, and overall execution time of the block in the runtime.

S. Transmitter Design

Both transmitters (secondary, primary) are designed and implemented in order to simulate the model described in (39) and (40). Fig. 13 depicts the block diagram of the secondary transmitter implemented in GNU Radio.

First a random byte generator source feeds the transmitting signal processing path with a pre-specified number of $J/8$ bytes. Then each byte is converted to data symbols of 8 bits. In that way, we come up with a sequence of J bits $\in \{0, 1\}$, where each of them is mapped to either 1 or -1 accordingly to the BPSK constellation; finally we get $b_k(i) \in \{-1, +1\}$. The generated bitstream is then given as an input to our own custom signal processing block which modulates each bit with a digital waveform of bits $s_k(l)$, $l = 0, \ldots L$. Moreover, the incoming stream is concatenated with a stream of pilot bits who play the role of a preamble and are exploited for time synchronization and phase estimation-correction at the receiver side. The preamble sequence is an input argument to our block, while the waveform-bit sequence is assumed to be known between every transmitter-receiver pair. Waveforms at the transmitter are dynamically changing according to the feedback information they get from the secondary receiver. Finally, pulse shaping is applied on the waveform-modulated bitstream. For this reason an already implemented GNU Radio block of a SRRC filter is used. The generated pulse train at the output of the last block provides the in-phase component of our IQ transmitter. The quadrature component is given by a null source.

In the case that $d_k(l)$, $k = 1, 2$, $l = 1, \ldots, L$ is fixed and not changing by sensing the feedback information of the secondary receiver the design proposed in Fig. 13 fits also the implementation of a primary transmitter.

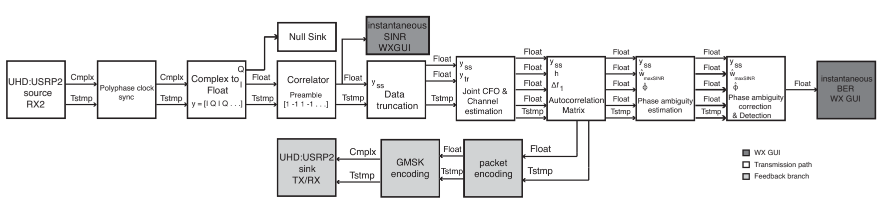


Fig. 14: Secondary receiver block diagram.

V. Conclusions

In this project, we proposed, developed and studied a decentralized algorithm for joint dynamic routing and code-division channelization in cognitive radio ad hoc networks. We considered the general problem of maximizing the network throughput through joint routing and spread-spectrum channelization. We proposed an algorithm that can be seen as a distributed localized approximation of the throughput-maximizing policy. The proposed algorithm requires solution of a code-division channelization problem as the search for the secondary amplitude, code transmission pair that maximizes the secondary link output SINR subject to the condition that all primary signal output SINR values are maintained above a given SINR-QoS threshold value. The formulated constrained optimization problem is non-convex and NP-hard in the code vector dimension. Simulation results show that the proposed algorithm considerably outperforms baseline solutions.

We also presented and evaluated through experiments RcUBe, a novel radio framework that offers abstractions to obtain real-time reconfigurability and self-optimization capabilities at the PHY, MAC, and network layers of the protocol stack. We demonstrated the flexibility of RcUBe by implementing a cross-layer cognitive radio algorithm designed to maximize the network throughput through jointly optimal control of routing, dynamic spectrum allocation, scheduling, transmit power control and interference avoiding waveform selection.

References

[1] I. F. Akyildiz,W. Lee, and K. Chowdhury, "CRAHNs: cognitive radio ad hoc networks," *Ad Hoc Networks Journal (Elsevier)*, vol. 7, pp. 810-836, Jul. 2009.

[2] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, pp. 201-220, Feb. 2005.

[3] H. Khalife, S. Ahuja, N. Malouch, and M. Krunz, "Probabilistic path selection in opportunistic cognitive radio networks", *Proc. of IEEE GLOBECOM*, New Orleans, LA, Dec. 2008.

[4] A. Abbagnale and F. Cuomo, "Connectivity-driven routing for cognitive radio ad-hoc networks", *Proc. of IEEE SECON*, Boston, MA, Jun. 2010.

[5] A. Abbagnale and F. Cuomo, "Leveraging the Algebraic Connectivity of a Cognitive Network for Routing Design", *IEEE Transactions on Mobile Computing*, vol. 11, no. 7, pp. 1163-1178, July 2012.

[6] I. Filippini, E. Ekici, and M.Cesana, "Minimum maintenance cost routing in cognitive radio networks", *Proc. of IEEE MASS*, Macau SAR, China, Oct. 2009.

[7] L. Ding, T. Melodia, S. Batalama, J. Matyjas, and M. Medley, "Cross-layer routing and dynamic spectrum allocation in cognitive radio ad hoc networks," *IEEE Trans. on Vehicular Technology*, vol. 59, no. 4, pp. 1969-1979, May 2010.

[8] R. Murawski and E. Ekici, "Utilizing dynamic spectrum leasing for cognitive radios in 802.11-based wireless networks," *Computer Networks Journal (Elsevier)*, vol. 55, no 11, pp. 2646-2657, Aug. 2011.

[9] K. R. Chowdhury and I. F. Akyildiz, "CRP: a routing protocol for cognitive radio ad hoc Networks", *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 29, no. 4, pp 794-804, Apr. 2011.

[10] Q. Zhu, Z. Yuan, J. B. Song, Z. Han, and T. Basar,"Interference aware routing game for cognitive radio multi-hop networks", *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 30, no. 10, pp. 2006-2015, Nov. 2012.

[11] M. Cesana, F. Cuomo, and E. Ekici, "Routing in cognitive radio networks: challenges and solutions," *Ad Hoc Networks Journal (Elsevier)*, vol. 9, no. 3, pp. 228-248, May 2011.

[12] Y. Xing, C. Mathur, M. A. Haleem, R. Chandramouli, and K. P. Subbalakshmi, "Dynamic spectrum access with QoS and interference temperature constraints," *IEEE Trans. Mobile Computing*, vol. 6, pp. 423-433, Apr. 2007.

[13] L. Qian, J. Attia, X. Li, and Z. Gajic, "Joint power control and admission control for CDMA cognitive radio networks," *in the 15th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN 2007)*, New Jersey, USA, Jun. 2007.

[14] L. B. Le and E. Hossain, "Resource allocation for spectrum underlay in cognitive radio networks,"*IEEE Trans. Wireless Commun.*, vol. 7, pp. 5306-5315, Dec. 2008.

[15] A. Elezabi, M. Kashef, M. Abdallah, M. Khairy, "CDMA underlay network with cognitive interference-minimizing code assignment and semi-blind interference suppression," in *Journ. Wireless Commun. Mob. Comput.*, vol. 9, pp. 1460-1471, 2009.

[16] M. Kashef, M. Abdallah, A. Elezabi, M. Khairy, "System parameter selection for asymmetric underlay CDMA networks with interference-minimizing code assignment," in *Proc. IEEE 10th Workshop on Signal Proc. Advances in Wireless Comm.*, Perugia, Italy, Jun. 2009.

[17] Q. Qu, L. B. Milstein, and D. R. Vaman, "Cognitive radio based multi-user resource allocation in mobile ad hoc networks using multi-carrier CDMA modulation," *IEEE J. Select. Areas Commun.*, vol. 26, pp. 70-82, Jan. 2008.

[18] M. H. Islam, Ying-chang Liang, Anh Hoang, "Joint beamforming and power control in the downlink of cognitive radio networks," *IEEE Tran. on Wireless Comm.*, vol. 7, no. 7, pp. 2415 2419, Jul. 2008.

[19] D. I. Kim, L. B. Le, and E. Hossain, "Joint rate and power allocation for cognitive radios in dynamic spectrum access environment, *IEEE Tran. on Wireless Comm.*, vol. 7, no. 12, pp.5517-5527, Dec. 2008.

[20] S. E. Safavi and K.P. Subbalakshmi, "Optimal joint power allocation and phase control for DS-CDMA cognitive radio networks, *in Proc. of IEEE Global telecom. conf. (GLOBECOM)*, Dec. 2011.

[21] B. Wang and D. Zhao, "Scheduling for long term proportional fairness in a cognitive wireless network with spectrum underlay," *IEEE Trans. on Wireless Comm.*, vol. 9, no. 3, pp. 1150- 1158, Mar. 2010.

[22] V. Chakravarthy, X. Li, Z. Wu, M. A. Temple, F. Garber, R. Kannan, A. Vasilakos, "Novel overlay/underlay cognitive radio waveforms using SD-SMSE framework to enhance spectrum efficiency - Part I: Theoretical framework and analysis in AWGN channel," *IEEE Trans. Commun.*, vol. 57, no. 12, pp. 3794-3804, Dec. 2009.

[23] F. Granelli, P. Pawelczak, R. V. Prasad, K. P. Subbalakshmi, R. Chandramouli, J. A. Hoffmeyer, and H. S. Berger, "Standardization and research in cognitive and dynamic spectrum access networks: IEEE SCC41 efforts

and other activities," *IEEE Commun. Magazine*, vol. 48, pp. 71-79, Jan. 2010.

[24] D. G. Manolakis, V. K. Ingle, S. M. Kogon, *Statistical and adaptive signal processing: spectral estimation, signal modeling, adaptive filtering, and array processing.* New York, NY: McGraw-Hill, 2000.

[25] J. G. Proakis, *Digital Communications.*Fourth Edition. McGraw-Hill Science, 2000.

[26] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling for maximum throughput in multihop radio networks," *IEEE Trans. on Automatic Control*, vol. 37, no. 12, pp. 1936-1949, 1992.

[27] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks", *Found. Trends Netw.*, vol. 1, no. 1, pp. 1-144, Apr. 2006.

[28] G. Sharma, N. B. Shroff, and R. R. Mazumdar, "On the complexity of scheduling in wireless networks", *In Proc. of ACM Intl. Conf. on Mobile Computing and Networking (MobiCom)*, Los Angeles, CA, USA, September 2006.

[29] P. M. Pardalos and S. A. Vavasis, "Quadratic programming with one negative eigenvalue is NP-hard," *J. Global Optim.*, vol. 1, pp. 15-22, 1991.

[30] M. Grant and S. Boyd, CVX Version 1.2–A MATLAB software, Feb. 2009.

[31] K. Gao, S. N. Batalama, D. A. Pados, and J. D. Matyjas, "Cognitive code-division channelization," *IEEE Trans. Wireless Commun.*, vol. 10, pp. 1090-1097, Apr. 2011.

[32] G. N. Karystinos and D. A. Pados, "New bounds on the total squared correlation and optimum design of DS-CDMA binary signature sets," *IEEE Trans. Commun.*, vol. 51, pp. 48-51, Jan. 2003.

[33] C. Ding, M. Golin, and T. Klϕve, "Meeting the Welch and Karystinos-Pados bounds on DS-CDMA binary signature sets," *Designs, Codes and Cryptography*, vol. 30, pp. 73-84, Aug. 2003.

[34] V. P. Ipatov, "On the Karystinos-Pados bounds and optimal binary DS-CDMA signature ensembles," *IEEE Commun. Lett.*, vol. 8, pp. 81-83, Feb. 2004.

[35] G. Nychis, T. Hottelier, Z. Yang, S. Seshan, and P. Steenkiste, "Enabling MAC protocol implementations on software-defined radios," in *Proceedings of USENIX Symp, on Networked systems design and implementation (NSDI)*, ser. NSDI'09, 2009, pp. 91–105.

[36] I. Tinnirello, G. Bianchi, P. Gallo, D. Garlisi, F. Giuliano, and F. Gringoli, "Wireless MAC processors: Programming MAC protocols on commodity Hardware," in *Proc. of IEEE INFOCOM*, Mar. 2012, pp. 1269 –1277.

[37] J. Tang and X. Zhang, "Cross-layer resource allocation over wireless relay networks for quality of service provisioning," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 4, pp. 645–656, 2007.

[38] P. Weeraddana, M. Codreanu, M. Latva-aho, and A. Ephremides, "Resource allocation for cross-layer utility maximization in wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 6, pp. 2790–2809, 2011.

[39] J. Liu, C. H. Xia, N. B. Shroff, and H. D. Sherali, "Distributed cross-layer optimization in wireless networks: A second-order approach," in *Proc. of IEEE INFOCOM*, 2013, pp. 2103–2111.

[40] Y. Shi, Y. Hou, J. Liu, and S. Kompella, "Bridging the gap between protocol and physical models for wireless networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 7, pp. 1404–1416, 2013.

[41] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks*, vol. 50, no. 13, pp. 2127–2159, 2006.

[42] H. Su and X. Zhang, "Cross-layer based opportunistic MAC protocols for QoS provisionings over cognitive radio wireless networks," *IEEE J. Select. Areas Commun.*, vol. 26, no. 1, pp. 118–129, 2008.

[43] L. Ding, T. Melodia, S. Batalama, J. Matyjas, and M. Medley, "Cross-layer routing and dynamic spectrum allocation in cognitive radio ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 4, pp. 1969–1979, May 2010.

[44] K. Gao, L. Ding, T. Melodia, S. Batalama, D. Pados, and J. Matyjas, "Spread-spectrum Cognitive Networking: Distributed Channelization and Routing," in *Proc. of IEEE Military Communications Conference (MILCOM)*, Baltimore, MD, November 2011.

[45] Z. Guan, T. Melodia, D. Yuan, and D. A. Pados, "Distributed Spectrum Management and Relay Selection in Interference-Limited Cooperative Wireless Networks," in *Proc. of ACM Intl. Conf. on Mobile Computing and Networking (MobiCom)*, Las Vegas, Nevada, USA, September 2011.

[46] S. Chen and A. M. Wyglinski, "Efficient spectrum utilization via cross-layer optimization in distributed cognitive radio networks," *Comput. Commun.*, vol. 32, no. 18, pp. 1931–1943, Dec. 2009.

[47] Y. Shi, Y. Hou, H. Zhou, and S. Midkiff, "Distributed cross-layer optimization for cognitive radio networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 8, pp. 4058–4069, 2010.

[48] Y. Song and J. Xie, "A distributed broadcast protocol in multi-hop cognitive radio ad hoc networks without a common control channel," in *Proc. of IEEE INFOCOM*, 2012, pp. 2273–2281.

[49] D. Xue and E. Ekici, "Cross-layer scheduling for cooperative multi-hop cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 3, pp. 534–543, 2013.

[50] M. Bansal, J. Mehlman, S. Katti, and P. Levis, "Openradio: a programmable wireless dataplane," in *Proc. of Workshop on Hot topics in software defined networks (HotSDN)*, ser. HotSDN '12, 2012, pp. 109–114.

[51] J. Ansari, X. Zhang, A. Achtzehn, M. Petrova, and P. Mahonen, "A flexible MAC development framework for cognitive radio systems," in *IEEE Conf. on Wireless Communications and Networking (WCNC)*, march 2011, pp. 156 –161.

[52] "USRP products," 2013, http://www.ettus.com. [Online]. Available: http://www.ettus.com

[53] "WARP," 2013, http://warp.rice.edu/trac.

[54] C. Doerr, M. Neufeld, J. Fifield, T. Weingart, D. Sicker, and D. Grunwald, "MultiMAC - an adaptive MAC framework for dynamic radio networking," in *IEEE Conf. on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, nov. 2005, pp. 548 –555.

[55] A. Sharma and E. M. Belding, "FreeMAC: framework for multi-channel MAC development on 802.11 hardware," in *Proc. of ACM Workshop on Programmable routers for extensible services of tomorrow (PRESTO)*, ser. PRESTO '08. New York, NY, USA: ACM, 2008, pp. 69–74.

[56] P. Djukic and P. Mohapatra, "Soft-TDMAC: A software tdma-based MAC over commodity 802.11 hardware," in *Proc. of IEEE INFOCOM*, april 2009, pp. 1836 –1844.

[57] "GNU Radio," 2013, http://gnuradio.org/redmine/projects/gnuradio/wiki.

[58] K. Tan, H. Liu, J. Zhang, Y. Zhang, J. Fang, and G. M. Voelker, "SORA: high-performance software radio using general-purpose multi-core processors," *Commun. ACM*, vol. 54, no. 1, pp. 99–107, Jan. 2011.

[59] M. C. Ng, K. Fleming, M. Vutukuru, S. Gross, A. Arvind, and H. Balakrishnan, "Airblue: A system for cross-layer wireless protocol development," in *ACM/IEEE Symp. on Architectures for Networking and Communications Systems (ANCS)*, oct. 2010, pp. 1 –11.

[60] G. Bianchi, P. Gallo, D. Garlisi, F. Giuliano, F. Gringoli, and I. Tinnirello, "MAClets: active MAC protocols over hard-coded devices," in *Proc. of Intl. Conf. on Emerging networking experiments and technologies (CoNEXT)*, ser. CoNEXT '12, 2012, pp. 229–240.

[61] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[62] K.-K. Yap, M. Kobayashi, R. Sherwood, T.-Y. Huang, M. Chan, N. Handigol, and N. McKeown, "Openroads: empowering research in mobile networks," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 1, pp. 125–126, Jan. 2010.

[63] K.-K. Yap, R. Sherwood, M. Kobayashi, T.-Y. Huang, M. Chan, N. Handigol, N. McKeown, and G. Parulkar, "Blueprint for introducing innovation into wireless mobile networks," in *Proc. of ACM Workshop on Virtualized infrastructure systems and architectures (VISA)*, 2010.

[64] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient collision-free medium access control for wireless sensor networks," in *Proceedings of Intl. Conf, on Embedded networked sensor systems (SenSys)*, ser. SenSys '03, 2003, pp. 181–192.

[65] K. Chowdhury and T. Melodia, "Platforms and testbeds for experimental evaluation of cognitive ad hoc networks," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 96–104, Sep. 2010.

[66] Connecting America: The Nation Broadband Plan (FCC), `http://download.broadband.gov/plan/national-broadband-plan.pdf`

[67] Federal Communications Commission, Spectrum Policy Task Force, Report ET Docket no. 02-135, Nov. 2002.

[68] I. F. Akyildiz, W. Lee, and K. Chowdhury, "CRAHNs: Cognitive Radio Ad Hoc Networks," *Ad Hoc Networks Journal (Elsevier),* vol. 7, pp. 810-836, Jul. 2009.

[69] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Select. Areas Commun.,* vol. 23, pp. 201-220, Feb. 2005.

[70] J. Mitola, "Cognitive Radio - An integrated agent architecture for software defined radio," *Ph.D. Dissertation*, Teleinformatics, Royal Institute of Technology, Sweden, 2000.

[71] K. Gao, S. N. Batalama, D. A. Pados, and J. D. Matyjas, "Cognitive CDMA channelization," in *Proceedings Forty Third Annual Conf. on Signals, Systems, and Computers, Pacific Grove, CA*, Nov. 1-4, 2009, pp. 672-676.

[72] M. Li, S. N. Batalama, D. A. Pados, T. Melodia, M. J. Medley and J. D. Matyjas, “Cognitive code-division channelization with blind primary-system identification,” in *Proceedings IEEE MILCOM 2010 (Conf. on Military Communication)*, San Jose, CA, Oct. 31-Nov. 3, 2010, pp. 1460-1465.

[73] K. Gao, S. N. Batalama, D. A. Pados, and J. D. Matyjas, “Cognitive code-division channelization,” in *IEEE Transactions on Wireless Communications,* vol. 10, no. 4, April 2011.

[74] K. Gao, L. Ding, T. Melodia, S. N. Batalama, D. A. Pados, and J. D. Matyjas, “Spread-spectrum cognitive networking: Distributed channelization and routing,” in *Proceedings IEEE MILCOM 2011 (Conf. on Military Communication),* Baltimore, MD, Nov. 7 - 10, 2011, pp. 1250-1255

[75] M. Li, S. N. Batalama, D. A. Pados, T. Melodia, M. J. Medley, and J. D. Matyjas, “Cognitive code-division links with blind primary-system identification, in *IEEE Transactions on Wireless Communications,* vol. 10, pp. 3743-3753, Nov. 2011.

[76] B. Wang and K. J. Ray Liu, “Advances in Cognitive Radio Networks: A Survey,” *IEEE J. on Sel. Topics in Sig. Proc.,* vol. 5, no. 1, Feb. 2011.

[77] M. Gkizeli, D. A. Pados, and M. J. Medley, “Optimal signature design for spread-spectrum steganography,” in *IEEE Transactions on Image Processing,* vol. 16, pp. 391-405, Feb. 2007.

[78] R. Zhou, Q. Han, R. Cooper, V. Chakravarthy, and Z. Wu, “A software defined radio based adaptive interference avoidance TDCS cognitive radio,” *IEEE International Conference on Communications (ICC),* pp. 1-5, 23-27 May 2010.

[79] R. Zhou, X. Li, V. Chakravarthy, C. Bullmaster, B. Wang, R. Cooper, and Z. Wu “Software defined radio implementation of SMSE based overlay cognitive radio,” *2010 IEEE Symposium on New Frontiers in Dynamic Spectrum,* pp. 1-2, 6-9 April 2010.

[80] G. N. Karystinos and D. A. Pados, “Supervised phase correction of blind space-time DS/CDMA channel estimates,” in *IEEE Transactions on Communications,* vol. 55, pp. 584-592, Mar. 2007.

[81] S. E. Bensley and B. Aazhang, “Subspace-based channel estimation for code division multiple access communication systems,” *IEEE Transactions on Communications*, vol. 44, no. 8, pp. 1009-1020, Aug. 1996.

[82] C. D. Richmond, “Derived PDF of maximum likelihood signal estimator which employs an estimated noise covariance,” in *IEEE Transactions on Signal Processing*, vol. 44, pp. 305-315, Feb. 1996.

[83] I. N. Psaromiligkos and S. N. Batalama, "Interference-plus-noise covariance matrix estimation for adaptive space-time processing of DS/CDMA signals," in *Proc. Vehicular Technology Conf. (VCT 2000)*, Boston, MA, Sep. 2000, vol. 5, pp. 2197-2204.

[84] P. B. Nagaraju, L. Ding, T. Melodia, S. N. Batalama, D. A. Pados, and J. D. Matyjas, "Implementation of a distributed joint routing and dynamic spectrum allocation algorithm on USRP2 radios," in *Proceedings 2010 IEEE Comm. Society Conf. on Sensor, Mesh and Ad Hoc Comm. and Networks (SECON)*, Boston, MA, June 21 - 25, 2010, pp. 1-2.

[85] S. Haykin, *Adaptive Filter Theory*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[86] H. L. Van Trees, *Detection Estimation and Modulation Theory Part I*, New York: Wiley, 2001.

[87] D. G. Manolakis, V. K. Ingle, and S. M. Kogon, *Statistical and Adaptive Signal Processing: Spectral Estimation, Signal Modeling, Adaptive Filtering and Array Processing,* McGraw-Hill, 2000.

[88] (2012) USRP products. Matt Ettus. [Online]. Available: `http://www.ettus.com`

[89] (2012) GNU radio. [Online]. Available: `http://gnuradio.org/redmine/projects/gnuradio/wiki`

[90] (2012) GNU radio, "GNUradio polyphase time synchronizer," accessed September 2012, `http://gnuradio.org/doc/doxygen/`

List of Acronyms

ADC	analog − to − digital converter
AWGN	additive white Gaussian noise
BER	bit − error − rate
BPSK	binary phase − shift − keying
CCC	common control channel
CSMA	carrier sense multiple access
CDMA	code − division multiple − access
CDM	code − division multiplexing
CFO	carrier frequency offset
CTS	Clear − to − Send
CR	cognitive radio
CRN	cognitive radio network
DAC	digital − to − analog converter
DC	data channel
DS − CDMA	direct − sequence code − division multiple − access
DTS	data transmission reservation
FDM	frequency − division multiplexing
FPGA	Field − Programmable Gate Arrays
ISI	inter − symbol − interference
I − SMIRA	interference constraint − aware stepwise maximum interference removal algorithm
LUT	look − up − table
MAC	medium − access − control
MC − CDMA	multi − carrier code − division multiple − access
MSE	mean − square − error
MSINR	maximum − SINR
OFDM	orthogonal frequency − division multiplexing
QCQP	quadratically constrained quadratic program
QPSK	quadrature phase − shift − keying
RTS	Request − to − Send
ROCH	Routing Code − division Channelization
RCA	random code assignment
RcUBe	Real − time Reconfigurable Radio
SDR	software defined radio
SINR	signal − to − interference − plus − noise ratio
SMI	sample − matrix − inversion method
SRRC	square − rootraisedcosine
TDMA	time − division multiple − access
USRP	Universal Software Radio Peripheral